Troels C. Petersen
Associate Professor, Niels Bohr Institute, Copenhagen, Denmark

NBI, 7th of June 2024

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 22th of May 2024,

in the graduate course:

Applied Machine Learning

at

Niels Bohr Institute, University of Copenhagen

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

Theodore Beevers

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

98

Troels C. Petersen
Course responsible

Applied Machine Learning initial project grading of Theodore Beevers

Submitted file structure:

```
description:
    /Users/thomasspieksma/Documents/Teaching/AppliedML2024/AppliedML2024_InstructorsFolder/Initial_Project_Thom
classification:
    NN-TensorFlow:
        /Users/thomasspieksma/Documents/Teaching/AppliedML2024/AppliedML2024_InstructorsFolder/Initial_Project_
        /Users/thomasspieksma/Documents/Teaching/AppliedML2024/AppliedML2024_InstructorsFolder/Initial_Project_
    XGBoost:
        /Users/thomasspieksma/Documents/Teaching/AppliedML2024/AppliedML2024_InstructorsFolder/Initial_Project_
        /Users/thomasspieksma/Documents/Teaching/AppliedML2024/AppliedML2024_InstructorsFolder/Initial_Project_
    MLPClassifier\_RandomSearchCV:
        /Users/thomasspieksma/Documents/Teaching/AppliedML2024/AppliedML2024_InstructorsFolder/Initial_Project_
        /Users/thomasspieksma/Documents/Teaching/AppliedML2024/AppliedML2024_InstructorsFolder/Initial_Project_
regression:
    RandomForestRegressor:
        /Users/thomasspieksma/Documents/Teaching/AppliedML2024/AppliedML2024_InstructorsFolder/Initial_Project_
        /Users/thomasspieksma/Documents/Teaching/AppliedML2024/AppliedML2024_InstructorsFolder/Initial_Project_
    NN-TensorFlow:
        /Users/thomasspieksma/Documents/Teaching/AppliedML2024/AppliedML2024_InstructorsFolder/Initial_Project_
        /Users/thomasspieksma/Documents/Teaching/AppliedML2024/AppliedML2024_InstructorsFolder/Initial_Project_
clustering:
    KMeans:
        /Users/thomasspieksma/Documents/Teaching/AppliedML2024/AppliedML2024_InstructorsFolder/Initial_Project_
        /Users/thomasspieksma/Documents/Teaching/AppliedML2024/AppliedML2024_InstructorsFolder/Initial_Project_
```

Best scoring classification: XGBoost with a cross entropy score of 0.0628

Best scoring regression: RandomForestRegressor with a relative MAD score of 2513.0

Best scoring clustering: KMeans with an accuracy score of 0.7521

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 $[0, 8]$:	5.5 points
Your solution entailed 3 different algorithms, which gives you a score of $[0, 6]$:	5.0 points
Your best performance for classification gave: $max(0, (-log(CrossEntropy - 0.01)) \times 2.6)$:	7.6 points
Your variable choice for classification was scored $4 \times (\sum VarFreq(you)/VarFreq(top))$:	2.6 points
Your classification had 0 penalties, totaling to:	0.0 points
Your best performance for regression gave: $max(0, (-log(MAD(\frac{E-T}{T}) - 3000) - 0.5) \times 4.6)$:	5.0 points
Your variable choice for regression was scored $5 \times (\sum VarFreq(you)/VarFreq(top))$:	4.5 points
Your regression had 0 penalties, totaling to:	0.0 points
Your best performance for clustering gave: $max(0, (Accuracy - 0.7) \times 16)$:	0.8 points
Your variable choice for clustering was scored $(\sum VarFreq(you)/VarFreq(top))$:	0.5 points
Your clustering had 0 penalties, totaling to:	0.0 points
Thus your total number of points was:	**98 points**

classification - NN-TensorFlow

The solution gave the following metrics:

Metric	Equation	Value
Accuracy	`sklearn.metrics.accuracy_score`	0.953733
AUC	`sklearn.metrics.auc`	0.983500
Cross entropy	`sklearn.metrics.log_loss`	0.124758

The solution produced the following plots:

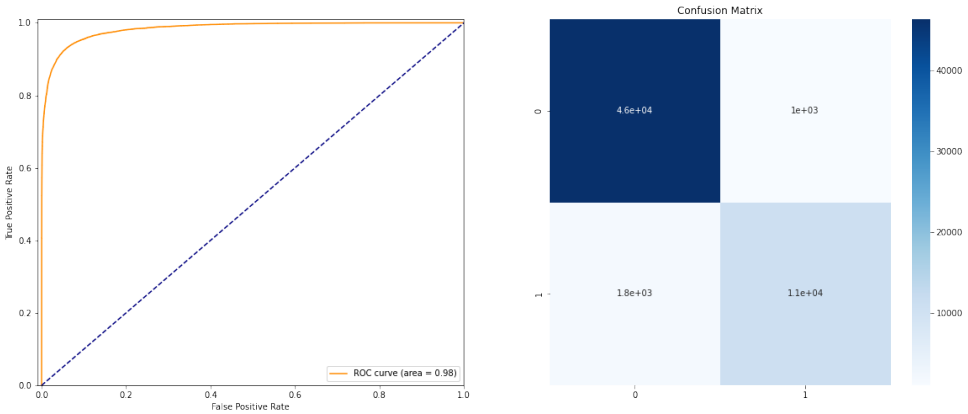

Figure 1: **Left:** ROC curve for the NN-TensorFlow implementation. The orange curve should be as close to the upper left corner as possible. **Right:** Confusion matrix for the NN-TensorFlow implementation. The diagonal squares ((0,0) and (1,1)) should have the higher values, compared to the squares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

These were the used variables:

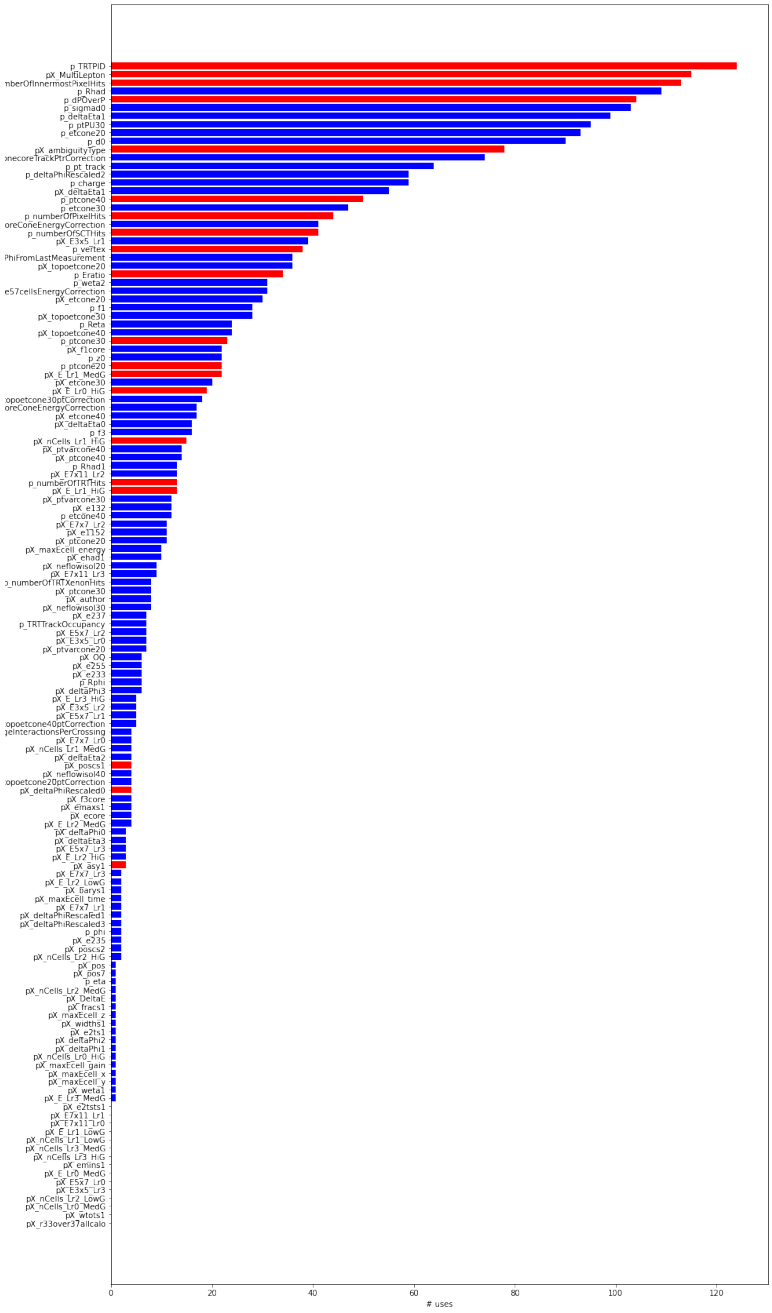

Figure 2: Shown in red are the variables that you used compared to those from your classmates.

classification - XGBoost

The solution gave the following metrics:

Metric	Equation	Value
Accuracy	`sklearn.metrics.accuracy_score`	0.977867
AUC	`sklearn.metrics.auc`	0.995244
Cross entropy	`sklearn.metrics.log_loss`	0.062820

The solution produced the following plots:

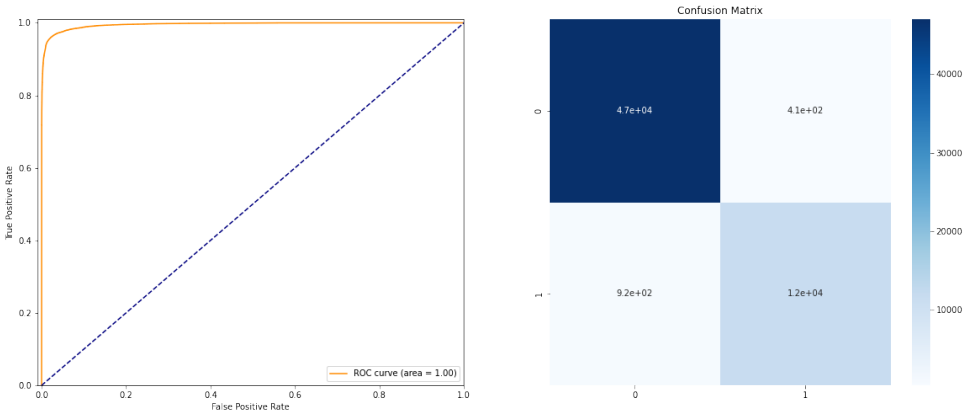

Figure 3: **Left:** ROC curve for the XGBoost implementation. The orange curve should be as close to the upper left corner as possible. **Right:** Confusion matrix for the XGBoost implementation. The diagonal squares ((0,0) and (1,1)) should have the higher values, compared to the squares in the other diagonal ((0,1) and (1,0)).

These were the used variables:

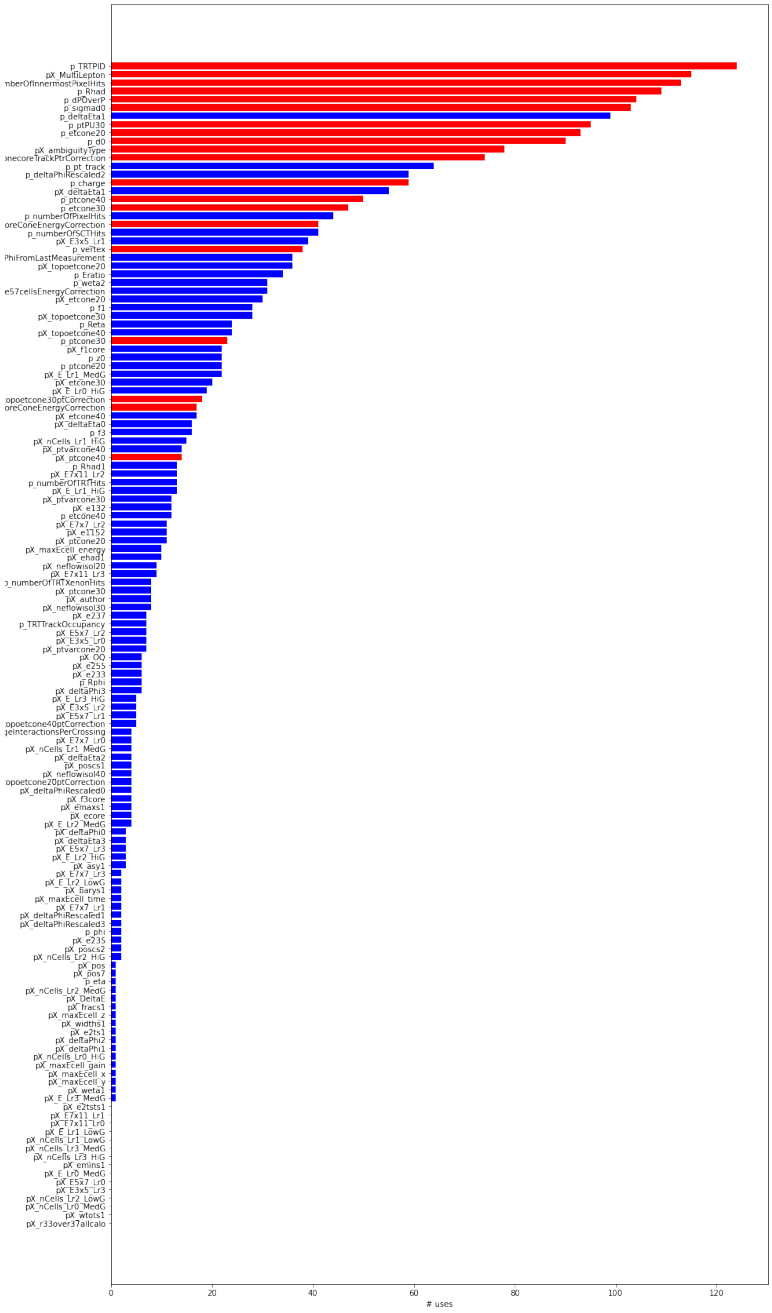

Figure 4: Shown in red are the variables that you used compared to those from your classmates.

classification - MLPClassifier_RandomSearchCV

The solution gave the following metrics:

Metric	Equation	Value
Accuracy	`sklearn.metrics.accuracy_score`	0.953167
AUC	`sklearn.metrics.auc`	0.983336
Cross entropy	`sklearn.metrics.log_loss`	0.124150

The solution produced the following plots:

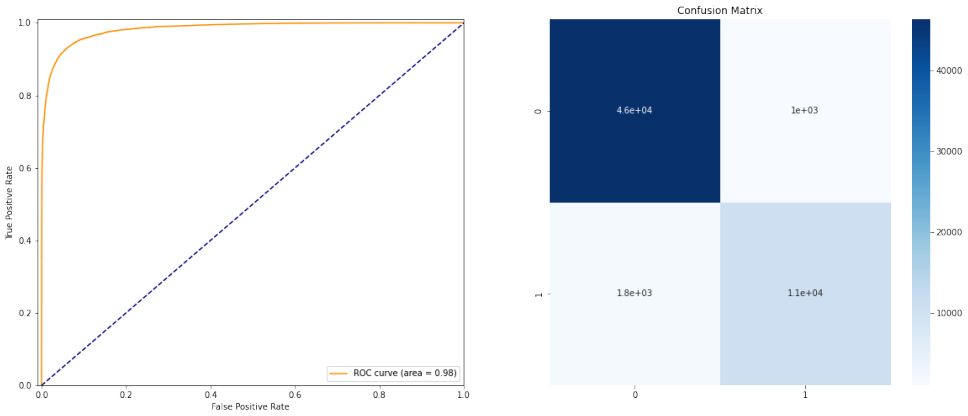

Figure 5: **Left:** ROC curve for the MLPClassifier_RandomSearchCV implementation. The orange curve should be as close to the upper left corner as possible. **Right:** Confusion matrix for the MLPClassifier_RandomSearchCV implementation. The diagonal squares ((0,0) and (1,1)) should have the higher values, compared to the squares in the other diagonal ((0,1) and (1,0)).

These were the used variables:

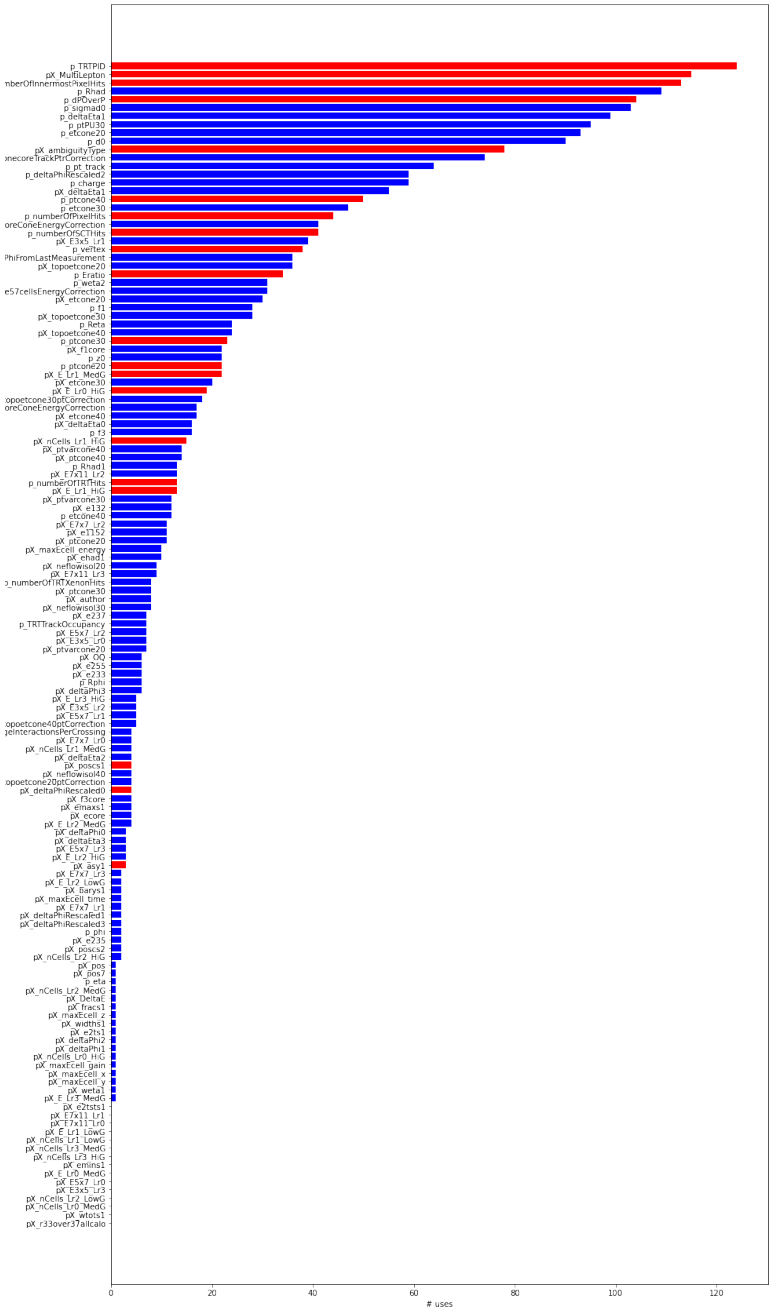

Figure 6: Shown in red are the variables that you used compared to those from your classmates.

regression - RandomForestRegressor

The solution gave the following metrics:

Metric	Equation	Value
MAE - Absolute	`sklearn.metrics.mean_absolute_error`	3081.8944
MAE - Relative	$\sum \lvert \frac{y_p - y_t}{y_t} \rvert$	2512.9569
RMS	$\sqrt{mean((y_p - y_t)^2)}$	6037.0018
RMS 98th percentile	$\sqrt{mean((y_p - y_t)^2)}$	4561.7830
RMS 90th percentile	$\sqrt{mean((y_p - y_t)^2)}$	3315.6943
RMS 70th percentile	$\sqrt{mean((y_p - y_t)^2)}$	2253.9471

The solution produced the following plots:

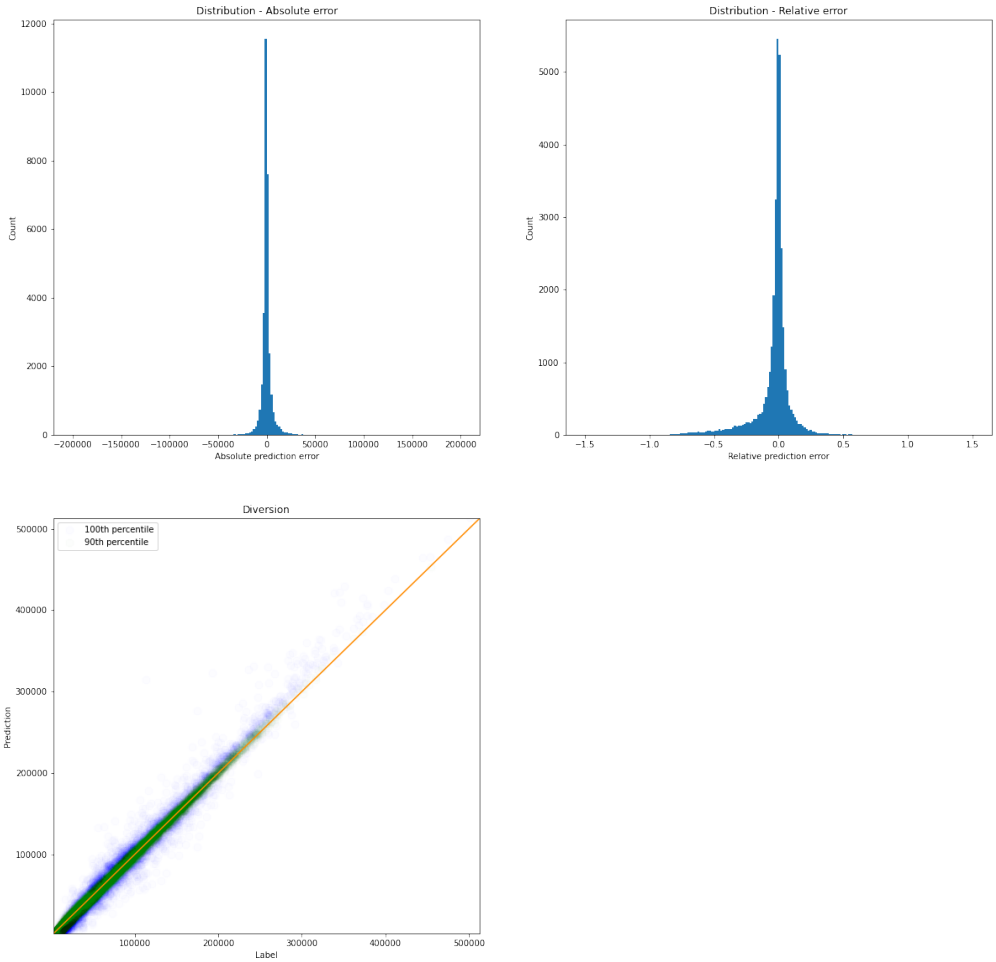

Figure 7: **Upper:** Distribution plots for the RandomForestRegressor implementation. The plots are for absolute error (*Left*) and relative error (*Right.*). Both plots should have a tall narrow curve, centered around 0. **Lower:** Diversion plot for the RandomForestRegressor implementation. The dots should be scattered close to the orange line - especially for the 90th percentile (green dots).

These were the used variables:

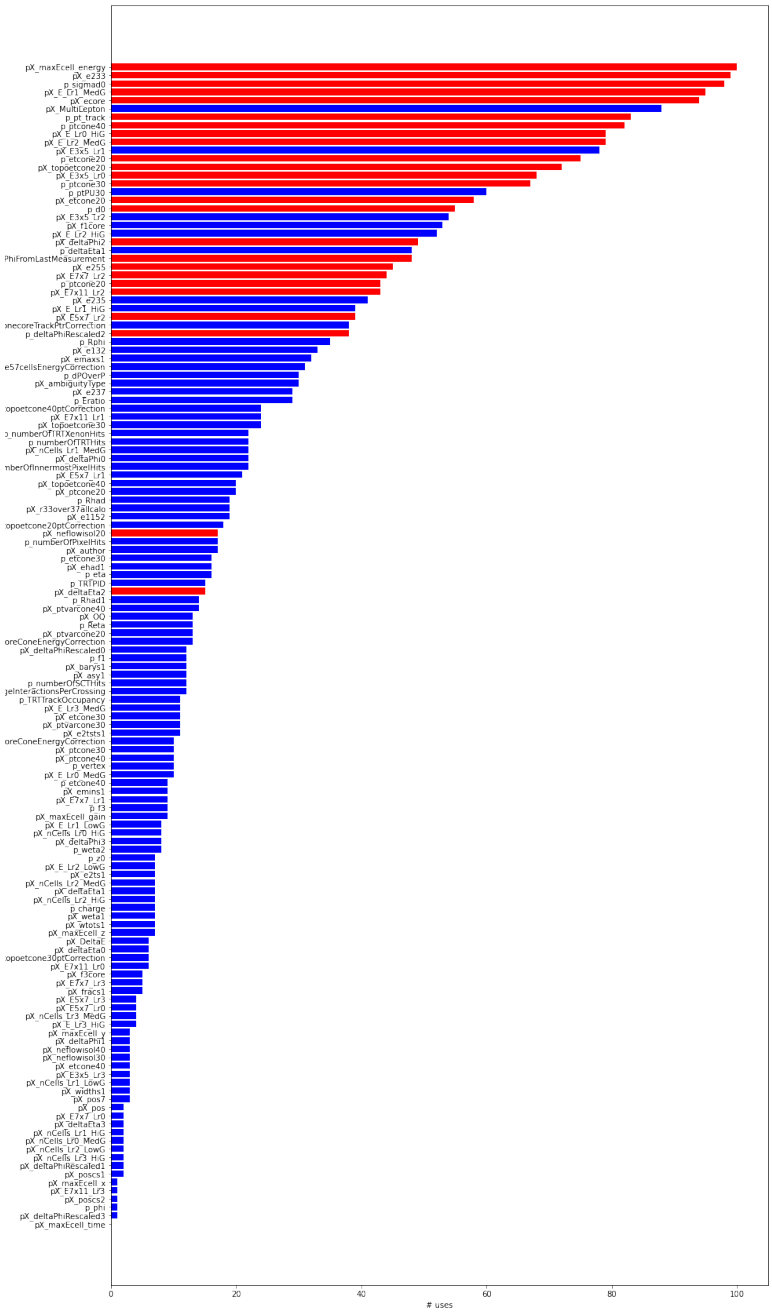

Figure 8: Shown in red are the variables that you used compared to those from your classmates.

regression - NN-TensorFlow

The solution gave the following metrics:

Metric	Equation	Value
MAE - Absolute	`sklearn.metrics.mean_absolute_error`	5245.3108
MAE - Relative	$\sum \lvert \frac{y_p - y_t}{y_t} \rvert$	3910.0121
RMS	$\sqrt{mean((y_p - y_t)^2)}$	10724.2773
RMS 98th percentile	$\sqrt{mean((y_p - y_t)^2)}$	7838.5951
RMS 90th percentile	$\sqrt{mean((y_p - y_t)^2)}$	5510.8894
RMS 70th percentile	$\sqrt{mean((y_p - y_t)^2)}$	3791.2431

The solution produced the following plots:

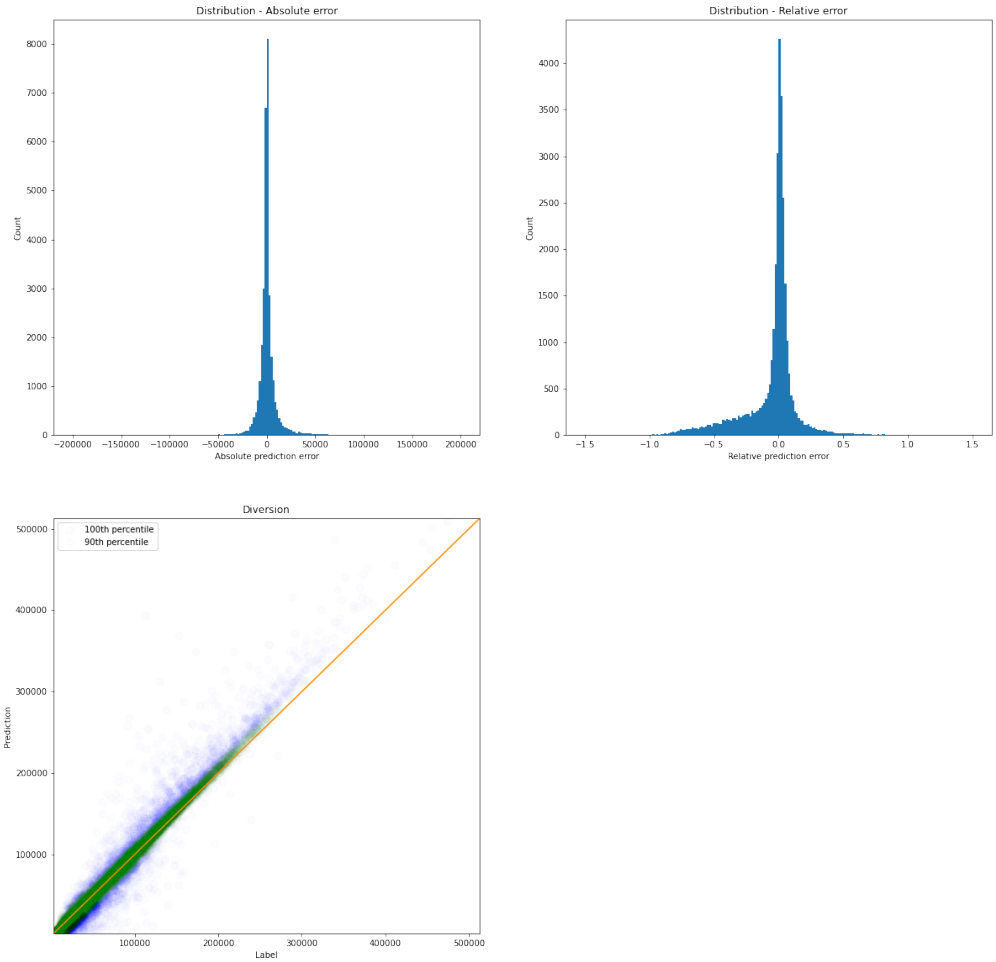

Figure 9: **Upper:** Distribution plots for the NN-TensorFlow implementation. The plots are for absolute error (*Left*) and relative error (*Right.*). Both plots should have a tall narrow curve, centered around 0. **Lower:** Diversion plot for the NN-TensorFlow implementation. The dots should be scattered close to the orange line - especially for the 90th percentile (green dots).

These were the used variables:

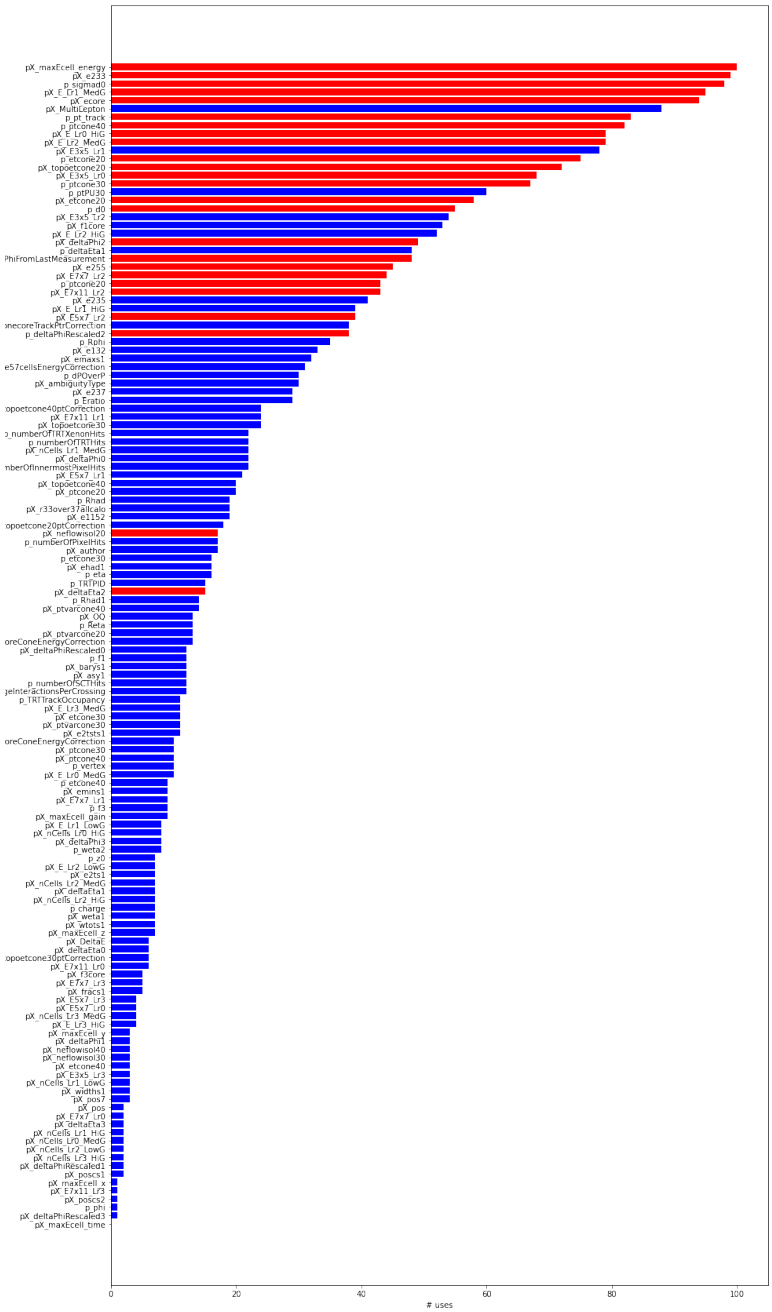

Figure 10: Shown in red are the variables that you used compared to those from your classmates.

clustering - KMeans

The solution produced the following metrics:

Metric	Equation	Value
Accuracy	`sklearn.metrics.accuracy_score`	0.7521

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
Type	2	2	2	2	2	2	2	0	2	2

The solution provided the following plot:

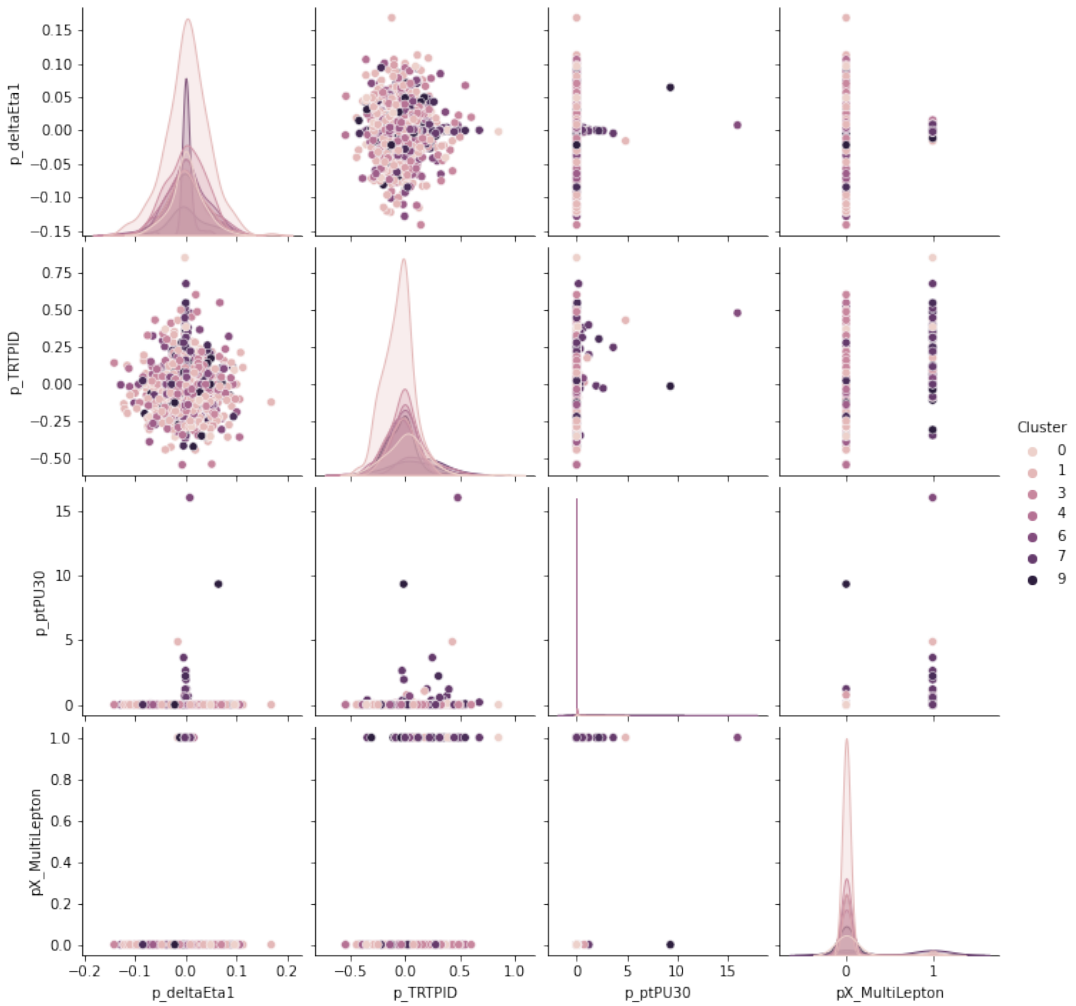

Figure 11: Pairplot for the KMeans implementation. The variables chosen are the top 4 most used variables for clustering. There should be a clear distinction of the clusters.

These were the used variables:

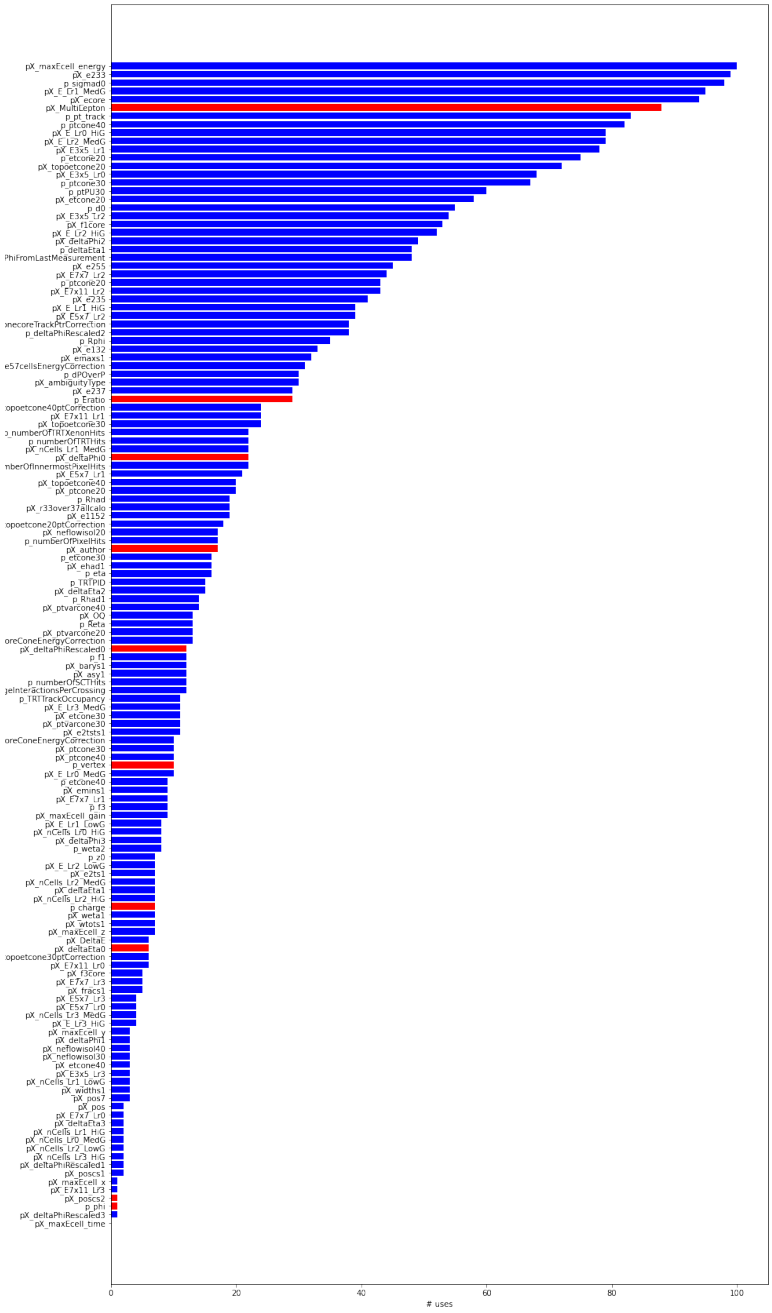

Figure 12: Shown in red are the variables that you used compared to those from your classmates.