Time	Group	Submission in Moodle; Mails with subject: [SMD2023]
Th. 12:00–13:00	A	lukas.beiske@udo.edu and tristan.gradetzke@udo.edu
Fr. 08:45–09:45	B	jonas.hackfeld@ruhr-uni-bochum.de and ludwig.neste@udo.edu
Fr. 10:00–11:00	C	stefan.froese@udo.edu and vincent.latko@udo.edu

Exercise 18 *Project Task: Feature Generation* **5 p.**

In this task, different types of particles ("tracks" and "cascades") are simulated and propagated through the detector. With the help of a random forest, a classification is performed. In order to do so, the simulated data must first be converted into a more useful form. Therefore, features are constructed from the deposited energies, which help with the classification. The Random Forest also calculates a "Feature Importance", which can be used to evaluate the importance of the features for the classification.

In this task, you will edit the method `analyse` of the `FeatureGenerator` class in `reconstruction/preprocessing/feature_generation.py`. To run and test your implementation, you can use the script `exercises/feature_generation.py` with the command:

```
$ # (replace <groupname> with the name of your group)
$ python exercises/feature_generation.py project_<groupname> -d plots/
```

If edited correctly, all tests should pass successfully. Keep in mind, however, that successfully passed tests do not equate to everything being correct!

Some features are already implemented to illustrate the procedure.

For testing purposes (especially on slower computers) it can be helpful to reduce the amount of simulated events. The default settings must be used for the final plot in your submission. The summary PDF created when testing your implementation will give you information about how suitable the features are for classification.

(a) Explain in general which features can be useful for a classification. In particular, address features that should not be used and discuss whether the features we provide are useful.

(b) Optimise the "roc auc score" by generating (at least three) new features. Use the available jupyter notebook (`notebooks/data_exploration.ipynb`) to get an overview of your features. Ideas for possible features can e.g. be found in the lecture. You will find the calculated "Feature Importances" in form of a box plot in the overview PDF.

(c) What is the highest "roc auc score" you have achieved? Take a look at the "Mean ROC Curve" graph in the overview PDF.

(d) What would a "roc auc score" of 0.5 and a "roc auc score" of 1 mean respectively?

(e) How could a "roc auc score" lower than 0.5 occur and give a solution to this problem.

(f) How do accuracy and sensitivity change with increasing "prediction threshold"? Take a look at the "Precision and Recall" graph in the overview PDF.

(g) Include the overview PDF in your submission.

Exercise 19 *Project Task: Energy Regression* **5 p.**

In addition to particle classification, energy regression also plays an important role in analyses in physics. Since the energy cannot be measured directly, machine learning methods are used to infer the energy of the particle with the help of measured quantities. The goal of this task is to use a random forest regressor and to perform a 5-fold cross validation. You will use the features that you implemented in the previous task.

In this task, the method will be added in `reconstruction/machine_learning/energy_regression`.

To run and test your implementation you can use the script `exercises/feature_generation.py` with the command:

```
$ # (replace <groupname> with the name of your group)
$ python exercises/energy_regression.py project_<groupname> -d plots/
```

The script is executable without any modifications, but the model predicts the mean energy of all particles for each event and no cross-validation is performed yet.

If edited correctly, all tests should pass successfully. Keep in mind, however, that successfully passed tests do not equate to everything being correct!

For testing purposes (especially on slower computers) it can be helpful to reduce the amount of simulated events. The default settings must be used for the final plot in your submission. The summary PDF created when testing your implementation will give you information about how well your energy regression is working.

(a) Change the function to use the random forest regressor in `define_model`. For this purpose, use the python package `sklearn`.

(b) Implement the 5-fold cross validation in the function `cross_validate_model`.

(c) You will find the graph "Regressor Confusion" in the overview PDF. Briefly describe what a migration matrix is. Which properties of your regressor can you infer from the migration matrix?

(d) Which properties of your regressor can you infer from the value "Bias" and "Resolution"? (Graph "Bias and Resolution")

(e) Which three features in this example are most important for the energy estimation? (Graph "Feature Importance")

(f) What is the effect of predicting the logarithmic energy instead of a linear prediction?

(g) Include the overview PDF in your submission.