CO2 Bericht

Author: Sebastian Pritz

Es wurden für die Ausarbeitung das Framework Sklearn (Python) mit einem Gradient-Boosted-Tree und Tensorflow (R) mit einem Neural Network ausgewählt. Letzteres hat nicht funktioniert, weswegen auf polynomielle Regression mit R umgestiegen wurde.

Allgemeines Datenpreprocessing

Das Datenpreprocessing wurde in Python, vorwiegend mithilfe von Pandas bewerkstelligt und befindet sich beiliegend im File „splitting.ipynb“.

Hierfür wurde die ursprüngliche Datei mittels Pandas hereingeladen und dann händisch mithilfe von Slicing in die verschiedenen Datensets laut PDF aufgeteilt.

Diese gefilterten Daten für Interpolation und Extrapolation wurden dann mithilfe der „index_number“ Spalte aus dem gesamten Datensatz entfernt, um disjunkte Trainings und Testdatensets zu erhalten.

Sklearn (Python)

Verwendet wurden hierbei die Packages pandas und sklearn.

Zu Beginn muss das Trainingsfile geladen, die „index_number“ Spalte gedropped, und das Datenset auf X und y Werte aufgeteilt werden.

Bezüglich Hyperparameter Optimierung entschloss ich mich im Falle des GBT aufgrund der vergleichsweise geringen Zeit zum Trainieren für einen GridSearch, welcher die Anzahl der Features, die Learning Rate, die Maximale Tiefe und die Anzahl der Bäume durchprobiert.

Dieser GridSearch wurde dann mithilfe der zuvor aufgeteilten Datensets mit einer k-fold-Cross-Validation mit k = 5 durchgeführt.

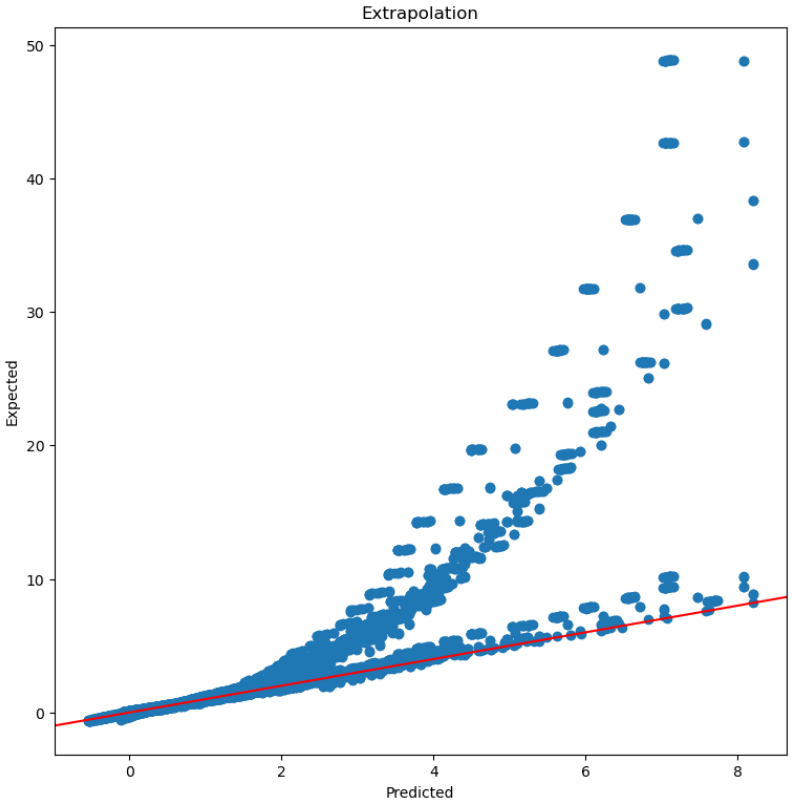
Die folgenden Werte wurden probiert:

```
parameters = {
              "criterion": ["friedman_mse"],
             "loss":["squared_error"],
             "max_features":["log2","sqrt"],
             'learning_rate': [0.01,0.1,0.5],
             'max_depth': [3, 4, 5, 6, 7],
             'n_estimators': [250, 500, 1000]
             }
```

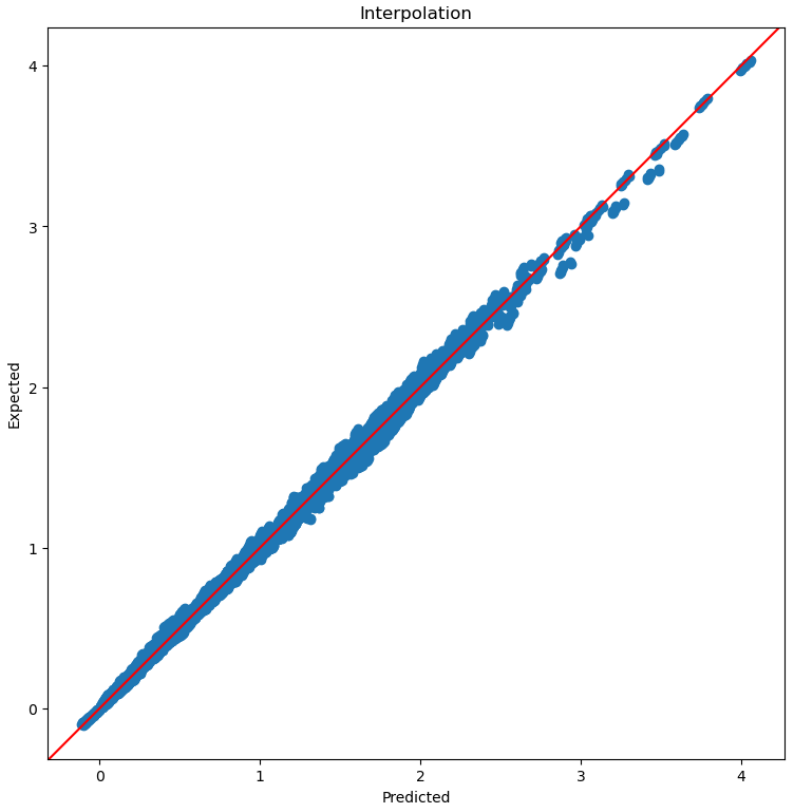
Anschließend können die Testdateien für Interpolation und Extrapolation geladen werden und die gewünschten Metriken MSE, MAE und R2 berechnet werden, indem das beste Modell des GridSearch verwendet wird, um die y-Werte vorherzusagen, welche dann mit den wirklichen Werten verglichen werden.

Die Interpolationsergebnisse (MSE: 0.00080, MAE: 0.02046, R2: 0.99757) fallen wie erwartet besser aus wie die Extrapolationsergebnisse (MSE: 3.80207, MAE: 0.29610, R2: 0.43667).

Wie man in untenstehender Visualisierung sehen kann, steigt der Fehler bei der Extrapolation nichtlinear mit dem erwarteten y.



Bei der Interpolation zeigen sich die schönen Ergebnisse auch in der Visualisierung.



Tensorflow (R)

Dieser Versuch wurde leider verworfen, da R aus welchen Gründen auch immer 22 GB RAM für die Modelldefinition (!) braucht, diese dann belegt, und dann erst recht abstürzt und den Speicher bis zum Neustart des Programms belegt.

R ist eine tolle Sprache mit super Frameworks /s

```
> model = neuralnet(
+   y~x1+x2+x3+x4,
+   data=train_data,
+   hidden=10,
+   err.fct = "sse",
+   learningrate = 0.003,
+   algorithm="backprop",
+   linear.output = TRUE
+ )
Error: cannot allocate vector of size 21.6 Gb
```

Polynomielle Regression (R)

Für diesen Ansatz wurden die Libraries Tidyverse, Caret und Metrics verwendet.

Für den Aufbau wurden wiederum alle Files eingelesen und dann nachträglich alle Spalten auf Numerics konvertiert.

Auch wenn sich aus zeitlichen Gründen keine Hyperparameteroptimierung ausging, wurde ein Validierungs Datenset gebildet und die Validation Accuracy bestimmt.

Als Modell wurde mithilfe der Formula „y“ abhängig von den anderen Variablen gemacht, wobei Polynome 5ten Grades verwendet wurden, was zeitlich leider das einzige Ergebnis war.

Interpolation (MSE: 0.037, MAE: 0.136, R2: 0.905)

Extrapolation (MSE: 5.298, MAE: 0.597, R2: 0.267)

Aufgrund von weiteren Problemen mit R und den poly/polym Libraries gingen sich leider dann auch keine Visualisierungen dazu aus.

Conclusio

Auch ohne Visualisierungen ist hier gerade, vielleicht wegen fehlendem Hyperparameter-Tuning, eindeutig dass das GBT-Modell in allen Metriken besser ist und der Polynomiellen Regression in R vorzuziehen ist, sowohl für Interpolation, als auch für Extrapolation. Die Qualität des GBT Modells, vor allem in Hinblick auf Interpolation ist sehr gut. Die der polynomiellen Regression, aus erwähnten Gründen, nicht bzw. weniger.