RandomForest_LOIC

June 29, 2021

[38]:

```
################################################################################
# Random Forest Classification Model (TensorFlow)                              #
# For LOIC Dataset                                                             #
# Based on the Implementation of:                                              #
# https://www.tensorflow.org/decision_forests/tutorials/beginner_colab         #
################################################################################
```

[39]:

```
# Installieren aller benötigten Pakete
!pip install numpy==1.19.2
!pip install six==1.15.0
!pip install wheel==0.35
!pip install tensorflow_decision_forests
!pip install pandas
!pip install wurlitzer
!pip install matplotlib
!pip install onnxruntime
!pip install keras2onnx
```

```
Requirement already satisfied: numpy==1.19.2 in
/home/julianbuecher/Projects/Bachelor-
Thesis/ML.Proxy.Python.ModelTrainer/lib/python3.8/site-packages (1.19.2)
Requirement already satisfied: six==1.15.0 in
/home/julianbuecher/Projects/Bachelor-
Thesis/ML.Proxy.Python.ModelTrainer/lib/python3.8/site-packages (1.15.0)
Requirement already satisfied: wheel==0.35 in
/home/julianbuecher/Projects/Bachelor-
Thesis/ML.Proxy.Python.ModelTrainer/lib/python3.8/site-packages (0.35.0)
Requirement already satisfied: packaging>=20.2 in
/home/julianbuecher/Projects/Bachelor-
Thesis/ML.Proxy.Python.ModelTrainer/lib/python3.8/site-packages (from
wheel==0.35) (20.9)
Requirement already satisfied: pyparsing>=2.0.2 in
/home/julianbuecher/Projects/Bachelor-
Thesis/ML.Proxy.Python.ModelTrainer/lib/python3.8/site-packages (from
packaging>=20.2->wheel==0.35) (2.4.7)
Requirement already satisfied: tensorflow_decision_forests in
/home/julianbuecher/Projects/Bachelor-
Thesis/ML.Proxy.Python.ModelTrainer/lib/python3.8/site-packages (0.1.7)
```

```
Requirement already satisfied: wheel in /home/julianbuecher/Projects/Bachelor-
Thesis/ML.Proxy.Python.ModelTrainer/lib/python3.8/site-packages (from
tensorflow_decision_forests) (0.35.0)
Requirement already satisfied: pandas in /home/julianbuecher/Projects/Bachelor-
Thesis/ML.Proxy.Python.ModelTrainer/lib/python3.8/site-packages (from
tensorflow_decision_forests) (1.2.5)
Requirement already satisfied: absl-py in /home/julianbuecher/Projects/Bachelor-
Thesis/ML.Proxy.Python.ModelTrainer/lib/python3.8/site-packages (from
tensorflow_decision_forests) (0.13.0)
Requirement already satisfied: six in /home/julianbuecher/Projects/Bachelor-
Thesis/ML.Proxy.Python.ModelTrainer/lib/python3.8/site-packages (from
tensorflow_decision_forests) (1.15.0)
Requirement already satisfied: numpy in /home/julianbuecher/Projects/Bachelor-
Thesis/ML.Proxy.Python.ModelTrainer/lib/python3.8/site-packages (from
tensorflow_decision_forests) (1.19.2)
Requirement already satisfied: tensorflow~=2.5 in
/home/julianbuecher/Projects/Bachelor-
Thesis/ML.Proxy.Python.ModelTrainer/lib/python3.8/site-packages (from
tensorflow_decision_forests) (2.5.0)
Requirement already satisfied: packaging>=20.2 in
/home/julianbuecher/Projects/Bachelor-
Thesis/ML.Proxy.Python.ModelTrainer/lib/python3.8/site-packages (from
wheel->tensorflow_decision_forests) (20.9)
Requirement already satisfied: python-dateutil>=2.7.3 in
/home/julianbuecher/Projects/Bachelor-
Thesis/ML.Proxy.Python.ModelTrainer/lib/python3.8/site-packages (from
pandas->tensorflow_decision_forests) (2.8.1)
Requirement already satisfied: pytz>=2017.3 in
/home/julianbuecher/Projects/Bachelor-
Thesis/ML.Proxy.Python.ModelTrainer/lib/python3.8/site-packages (from
pandas->tensorflow_decision_forests) (2021.1)
Requirement already satisfied: h5py~=3.1.0 in
/home/julianbuecher/Projects/Bachelor-
Thesis/ML.Proxy.Python.ModelTrainer/lib/python3.8/site-packages (from
tensorflow~=2.5->tensorflow_decision_forests) (3.1.0)
Requirement already satisfied: termcolor~=1.1.0 in
/home/julianbuecher/Projects/Bachelor-
Thesis/ML.Proxy.Python.ModelTrainer/lib/python3.8/site-packages (from
tensorflow~=2.5->tensorflow_decision_forests) (1.1.0)
Requirement already satisfied: keras-nightly~=2.5.0.dev in
/home/julianbuecher/Projects/Bachelor-
Thesis/ML.Proxy.Python.ModelTrainer/lib/python3.8/site-packages (from
tensorflow~=2.5->tensorflow_decision_forests) (2.5.0.dev2021032900)
Requirement already satisfied: flatbuffers~=1.12.0 in
/home/julianbuecher/Projects/Bachelor-
Thesis/ML.Proxy.Python.ModelTrainer/lib/python3.8/site-packages (from
tensorflow~=2.5->tensorflow_decision_forests) (1.12)
Requirement already satisfied: tensorboard~=2.5 in
```

```
/home/julianbuecher/Projects/Bachelor-
Thesis/ML.Proxy.Python.ModelTrainer/lib/python3.8/site-packages (from
tensorflow~=2.5->tensorflow_decision_forests) (2.5.0)
Requirement already satisfied: keras-preprocessing~=1.1.2 in
/home/julianbuecher/Projects/Bachelor-
Thesis/ML.Proxy.Python.ModelTrainer/lib/python3.8/site-packages (from
tensorflow~=2.5->tensorflow_decision_forests) (1.1.2)
Requirement already satisfied: astunparse~=1.6.3 in
/home/julianbuecher/Projects/Bachelor-
Thesis/ML.Proxy.Python.ModelTrainer/lib/python3.8/site-packages (from
tensorflow~=2.5->tensorflow_decision_forests) (1.6.3)
Requirement already satisfied: google-pasta~=0.2 in
/home/julianbuecher/Projects/Bachelor-
Thesis/ML.Proxy.Python.ModelTrainer/lib/python3.8/site-packages (from
tensorflow~=2.5->tensorflow_decision_forests) (0.2.0)
Requirement already satisfied: wrapt~=1.12.1 in
/home/julianbuecher/Projects/Bachelor-
Thesis/ML.Proxy.Python.ModelTrainer/lib/python3.8/site-packages (from
tensorflow~=2.5->tensorflow_decision_forests) (1.12.1)
Requirement already satisfied: grpcio~=1.34.0 in
/home/julianbuecher/Projects/Bachelor-
Thesis/ML.Proxy.Python.ModelTrainer/lib/python3.8/site-packages (from
tensorflow~=2.5->tensorflow_decision_forests) (1.34.1)
Requirement already satisfied: typing-extensions~=3.7.4 in
/home/julianbuecher/Projects/Bachelor-
Thesis/ML.Proxy.Python.ModelTrainer/lib/python3.8/site-packages (from
tensorflow~=2.5->tensorflow_decision_forests) (3.7.4.3)
Requirement already satisfied: opt-einsum~=3.3.0 in
/home/julianbuecher/Projects/Bachelor-
Thesis/ML.Proxy.Python.ModelTrainer/lib/python3.8/site-packages (from
tensorflow~=2.5->tensorflow_decision_forests) (3.3.0)
Requirement already satisfied: tensorflow-estimator<2.6.0,>=2.5.0rc0 in
/home/julianbuecher/Projects/Bachelor-
Thesis/ML.Proxy.Python.ModelTrainer/lib/python3.8/site-packages (from
tensorflow~=2.5->tensorflow_decision_forests) (2.5.0)
Requirement already satisfied: protobuf>=3.9.2 in
/home/julianbuecher/Projects/Bachelor-
Thesis/ML.Proxy.Python.ModelTrainer/lib/python3.8/site-packages (from
tensorflow~=2.5->tensorflow_decision_forests) (3.17.3)
Requirement already satisfied: gast==0.4.0 in
/home/julianbuecher/Projects/Bachelor-
Thesis/ML.Proxy.Python.ModelTrainer/lib/python3.8/site-packages (from
tensorflow~=2.5->tensorflow_decision_forests) (0.4.0)
Requirement already satisfied: pyparsing>=2.0.2 in
/home/julianbuecher/Projects/Bachelor-
Thesis/ML.Proxy.Python.ModelTrainer/lib/python3.8/site-packages (from
packaging>=20.2->wheel->tensorflow_decision_forests) (2.4.7)
Requirement already satisfied: markdown>=2.6.8 in
```

```
/home/julianbuecher/Projects/Bachelor-
Thesis/ML.Proxy.Python.ModelTrainer/lib/python3.8/site-packages (from
tensorboard~=2.5->tensorflow~=2.5->tensorflow_decision_forests) (3.3.4)
Requirement already satisfied: tensorboard-data-server<0.7.0,>=0.6.0 in
/home/julianbuecher/Projects/Bachelor-
Thesis/ML.Proxy.Python.ModelTrainer/lib/python3.8/site-packages (from
tensorboard~=2.5->tensorflow~=2.5->tensorflow_decision_forests) (0.6.1)
Requirement already satisfied: tensorboard-plugin-wit>=1.6.0 in
/home/julianbuecher/Projects/Bachelor-
Thesis/ML.Proxy.Python.ModelTrainer/lib/python3.8/site-packages (from
tensorboard~=2.5->tensorflow~=2.5->tensorflow_decision_forests) (1.8.0)
Requirement already satisfied: setuptools>=41.0.0 in
/home/julianbuecher/Projects/Bachelor-
Thesis/ML.Proxy.Python.ModelTrainer/lib/python3.8/site-packages (from
tensorboard~=2.5->tensorflow~=2.5->tensorflow_decision_forests) (44.0.0)
Requirement already satisfied: google-auth<2,>=1.6.3 in
/home/julianbuecher/Projects/Bachelor-
Thesis/ML.Proxy.Python.ModelTrainer/lib/python3.8/site-packages (from
tensorboard~=2.5->tensorflow~=2.5->tensorflow_decision_forests) (1.32.0)
Requirement already satisfied: werkzeug>=0.11.15 in
/home/julianbuecher/Projects/Bachelor-
Thesis/ML.Proxy.Python.ModelTrainer/lib/python3.8/site-packages (from
tensorboard~=2.5->tensorflow~=2.5->tensorflow_decision_forests) (2.0.1)
Requirement already satisfied: google-auth-oauthlib<0.5,>=0.4.1 in
/home/julianbuecher/Projects/Bachelor-
Thesis/ML.Proxy.Python.ModelTrainer/lib/python3.8/site-packages (from
tensorboard~=2.5->tensorflow~=2.5->tensorflow_decision_forests) (0.4.4)
Requirement already satisfied: requests<3,>=2.21.0 in
/home/julianbuecher/Projects/Bachelor-
Thesis/ML.Proxy.Python.ModelTrainer/lib/python3.8/site-packages (from
tensorboard~=2.5->tensorflow~=2.5->tensorflow_decision_forests) (2.25.1)
Requirement already satisfied: rsa<5,>=3.1.4; python_version >= "3.6" in
/home/julianbuecher/Projects/Bachelor-
Thesis/ML.Proxy.Python.ModelTrainer/lib/python3.8/site-packages (from google-
auth<2,>=1.6.3->tensorboard~=2.5->tensorflow~=2.5->tensorflow_decision_forests)
(4.7.2)
Requirement already satisfied: pyasn1-modules>=0.2.1 in
/home/julianbuecher/Projects/Bachelor-
Thesis/ML.Proxy.Python.ModelTrainer/lib/python3.8/site-packages (from google-
auth<2,>=1.6.3->tensorboard~=2.5->tensorflow~=2.5->tensorflow_decision_forests)
(0.2.8)
Requirement already satisfied: cachetools<5.0,>=2.0.0 in
/home/julianbuecher/Projects/Bachelor-
Thesis/ML.Proxy.Python.ModelTrainer/lib/python3.8/site-packages (from google-
auth<2,>=1.6.3->tensorboard~=2.5->tensorflow~=2.5->tensorflow_decision_forests)
(4.2.2)
Requirement already satisfied: requests-oauthlib>=0.7.0 in
/home/julianbuecher/Projects/Bachelor-
```

```
Thesis/ML.Proxy.Python.ModelTrainer/lib/python3.8/site-packages (from google-
auth-oauthlib<0.5,>=0.4.1->tensorboard~=2.5->tensorflow~=2.5->tensorflow_decisio
n_forests) (1.3.0)
Requirement already satisfied: certifi>=2017.4.17 in
/home/julianbuecher/Projects/Bachelor-
Thesis/ML.Proxy.Python.ModelTrainer/lib/python3.8/site-packages (from requests<3
,>=2.21.0->tensorboard~=2.5->tensorflow~=2.5->tensorflow_decision_forests)
(2021.5.30)
Requirement already satisfied: idna<3,>=2.5 in
/home/julianbuecher/Projects/Bachelor-
Thesis/ML.Proxy.Python.ModelTrainer/lib/python3.8/site-packages (from requests<3
,>=2.21.0->tensorboard~=2.5->tensorflow~=2.5->tensorflow_decision_forests)
(2.10)
Requirement already satisfied: urllib3<1.27,>=1.21.1 in
/home/julianbuecher/Projects/Bachelor-
Thesis/ML.Proxy.Python.ModelTrainer/lib/python3.8/site-packages (from requests<3
,>=2.21.0->tensorboard~=2.5->tensorflow~=2.5->tensorflow_decision_forests)
(1.26.6)
Requirement already satisfied: chardet<5,>=3.0.2 in
/home/julianbuecher/Projects/Bachelor-
Thesis/ML.Proxy.Python.ModelTrainer/lib/python3.8/site-packages (from requests<3
,>=2.21.0->tensorboard~=2.5->tensorflow~=2.5->tensorflow_decision_forests)
(4.0.0)
Requirement already satisfied: pyasn1>=0.1.3 in
/home/julianbuecher/Projects/Bachelor-
Thesis/ML.Proxy.Python.ModelTrainer/lib/python3.8/site-packages (from
rsa<5,>=3.1.4; python_version >= "3.6"->google-
auth<2,>=1.6.3->tensorboard~=2.5->tensorflow~=2.5->tensorflow_decision_forests)
(0.4.8)
Requirement already satisfied: oauthlib>=3.0.0 in
/home/julianbuecher/Projects/Bachelor-
Thesis/ML.Proxy.Python.ModelTrainer/lib/python3.8/site-packages (from requests-
oauthlib>=0.7.0->google-auth-oauthlib<0.5,>=0.4.1->tensorboard~=2.5->tensorflow~
=2.5->tensorflow_decision_forests) (3.1.1)
Requirement already satisfied: pandas in /home/julianbuecher/Projects/Bachelor-
Thesis/ML.Proxy.Python.ModelTrainer/lib/python3.8/site-packages (1.2.5)
Requirement already satisfied: python-dateutil>=2.7.3 in
/home/julianbuecher/Projects/Bachelor-
Thesis/ML.Proxy.Python.ModelTrainer/lib/python3.8/site-packages (from pandas)
(2.8.1)
Requirement already satisfied: pytz>=2017.3 in
/home/julianbuecher/Projects/Bachelor-
Thesis/ML.Proxy.Python.ModelTrainer/lib/python3.8/site-packages (from pandas)
(2021.1)
Requirement already satisfied: numpy>=1.16.5 in
/home/julianbuecher/Projects/Bachelor-
Thesis/ML.Proxy.Python.ModelTrainer/lib/python3.8/site-packages (from pandas)
(1.19.2)
```

```
Requirement already satisfied: six>=1.5 in
/home/julianbuecher/Projects/Bachelor-
Thesis/ML.Proxy.Python.ModelTrainer/lib/python3.8/site-packages (from python-
dateutil>=2.7.3->pandas) (1.15.0)
Requirement already satisfied: wurlitzer in
/home/julianbuecher/Projects/Bachelor-
Thesis/ML.Proxy.Python.ModelTrainer/lib/python3.8/site-packages (2.1.0)
Requirement already satisfied: matplotlib in
/home/julianbuecher/Projects/Bachelor-
Thesis/ML.Proxy.Python.ModelTrainer/lib/python3.8/site-packages (3.4.2)
Requirement already satisfied: pillow>=6.2.0 in
/home/julianbuecher/Projects/Bachelor-
Thesis/ML.Proxy.Python.ModelTrainer/lib/python3.8/site-packages (from
matplotlib) (8.2.0)
Requirement already satisfied: pyparsing>=2.2.1 in
/home/julianbuecher/Projects/Bachelor-
Thesis/ML.Proxy.Python.ModelTrainer/lib/python3.8/site-packages (from
matplotlib) (2.4.7)
Requirement already satisfied: cycler>=0.10 in
/home/julianbuecher/Projects/Bachelor-
Thesis/ML.Proxy.Python.ModelTrainer/lib/python3.8/site-packages (from
matplotlib) (0.10.0)
Requirement already satisfied: numpy>=1.16 in
/home/julianbuecher/Projects/Bachelor-
Thesis/ML.Proxy.Python.ModelTrainer/lib/python3.8/site-packages (from
matplotlib) (1.19.2)
Requirement already satisfied: python-dateutil>=2.7 in
/home/julianbuecher/Projects/Bachelor-
Thesis/ML.Proxy.Python.ModelTrainer/lib/python3.8/site-packages (from
matplotlib) (2.8.1)
Requirement already satisfied: kiwisolver>=1.0.1 in
/home/julianbuecher/Projects/Bachelor-
Thesis/ML.Proxy.Python.ModelTrainer/lib/python3.8/site-packages (from
matplotlib) (1.3.1)
Requirement already satisfied: six in /home/julianbuecher/Projects/Bachelor-
Thesis/ML.Proxy.Python.ModelTrainer/lib/python3.8/site-packages (from
cycler>=0.10->matplotlib) (1.15.0)
Requirement already satisfied: onnxruntime in
/home/julianbuecher/Projects/Bachelor-
Thesis/ML.Proxy.Python.ModelTrainer/lib/python3.8/site-packages (1.8.0)
Requirement already satisfied: protobuf in
/home/julianbuecher/Projects/Bachelor-
Thesis/ML.Proxy.Python.ModelTrainer/lib/python3.8/site-packages (from
onnxruntime) (3.17.3)
Requirement already satisfied: numpy>=1.16.6 in
/home/julianbuecher/Projects/Bachelor-
Thesis/ML.Proxy.Python.ModelTrainer/lib/python3.8/site-packages (from
onnxruntime) (1.19.2)
```

```
Requirement already satisfied: flatbuffers in
/home/julianbuecher/Projects/Bachelor-
Thesis/ML.Proxy.Python.ModelTrainer/lib/python3.8/site-packages (from
onnxruntime) (1.12)
Requirement already satisfied: six>=1.9 in
/home/julianbuecher/Projects/Bachelor-
Thesis/ML.Proxy.Python.ModelTrainer/lib/python3.8/site-packages (from
protobuf->onnxruntime) (1.15.0)
Requirement already satisfied: keras2onnx in
/home/julianbuecher/Projects/Bachelor-
Thesis/ML.Proxy.Python.ModelTrainer/lib/python3.8/site-packages (1.7.0)
Requirement already satisfied: numpy in /home/julianbuecher/Projects/Bachelor-
Thesis/ML.Proxy.Python.ModelTrainer/lib/python3.8/site-packages (from
keras2onnx) (1.19.2)
Requirement already satisfied: protobuf in
/home/julianbuecher/Projects/Bachelor-
Thesis/ML.Proxy.Python.ModelTrainer/lib/python3.8/site-packages (from
keras2onnx) (3.17.3)
Requirement already satisfied: fire in /home/julianbuecher/Projects/Bachelor-
Thesis/ML.Proxy.Python.ModelTrainer/lib/python3.8/site-packages (from
keras2onnx) (0.4.0)
Requirement already satisfied: requests in
/home/julianbuecher/Projects/Bachelor-
Thesis/ML.Proxy.Python.ModelTrainer/lib/python3.8/site-packages (from
keras2onnx) (2.25.1)
Requirement already satisfied: onnx in /home/julianbuecher/Projects/Bachelor-
Thesis/ML.Proxy.Python.ModelTrainer/lib/python3.8/site-packages (from
keras2onnx) (1.9.0)
Requirement already satisfied: onnxconverter-common>=1.7.0 in
/home/julianbuecher/Projects/Bachelor-
Thesis/ML.Proxy.Python.ModelTrainer/lib/python3.8/site-packages (from
keras2onnx) (1.8.1)
Requirement already satisfied: six>=1.9 in
/home/julianbuecher/Projects/Bachelor-
Thesis/ML.Proxy.Python.ModelTrainer/lib/python3.8/site-packages (from
protobuf->keras2onnx) (1.15.0)
Requirement already satisfied: termcolor in
/home/julianbuecher/Projects/Bachelor-
Thesis/ML.Proxy.Python.ModelTrainer/lib/python3.8/site-packages (from
fire->keras2onnx) (1.1.0)
Requirement already satisfied: urllib3<1.27,>=1.21.1 in
/home/julianbuecher/Projects/Bachelor-
Thesis/ML.Proxy.Python.ModelTrainer/lib/python3.8/site-packages (from
requests->keras2onnx) (1.26.6)
Requirement already satisfied: chardet<5,>=3.0.2 in
/home/julianbuecher/Projects/Bachelor-
Thesis/ML.Proxy.Python.ModelTrainer/lib/python3.8/site-packages (from
requests->keras2onnx) (4.0.0)
```

```
Requirement already satisfied: idna<3,>=2.5 in
/home/julianbuecher/Projects/Bachelor-
Thesis/ML.Proxy.Python.ModelTrainer/lib/python3.8/site-packages (from
requests->keras2onnx) (2.10)
Requirement already satisfied: certifi>=2017.4.17 in
/home/julianbuecher/Projects/Bachelor-
Thesis/ML.Proxy.Python.ModelTrainer/lib/python3.8/site-packages (from
requests->keras2onnx) (2021.5.30)
Requirement already satisfied: typing-extensions>=3.6.2.1 in
/home/julianbuecher/Projects/Bachelor-
Thesis/ML.Proxy.Python.ModelTrainer/lib/python3.8/site-packages (from
onnx->keras2onnx) (3.7.4.3)
```

[40]:

```python
# Laden der benötigten Python Pakete
import os
# os.environ["TF_KERAS"]='1'
import pandas as pd
import numpy as np
import tensorflow_decision_forests as tfdf
import tensorflow as tf
from wurlitzer import sys_pipes
import matplotlib.pyplot as plt
import onnx
import keras2onnx as k2o
```

[41]:

```python
# Prüfung der installierten TensorFlow Decision Forests Version
print(f"Found TensorFlow Decision Forests v{tfdf.__version__}")
```

```
Found TensorFlow Decision Forests v0.1.7
```

[42]:

```python
# Laden der Netzwerk Traffic Daten für den LOIC Angriff
data_LOIC = pd.read_csv('../Data/Tuesday-20-02-2018_LOIC-Attack.csv')
# Umbenennen der Spalten für bessere Kompatibiltät mit TensorFlow
data_LOIC.rename(columns={
    'Bwd Pkt Len Std':'bwd_pkt_len_std',
    'Pkt Size Avg':'pkt_size_avg',
    'Flow Duration':'flow_duration',
    'Flow IAT Std':'flow_iat_std',
    'Label':'label'},
    inplace=True)
```

[43]:

```python
# Festlegen der Label-Spalte innerhalb des Datasets
label = 'label'
```

[44]:

```python
# Aufteilen des Datasets in Training- und Test-Daten
def split_dataset(dataset,  test_ratio=0.30):
    """Splits a panda dataframe in two dataframes."""
    test_indices = np.random.rand(len(dataset)) < test_ratio
    return dataset[~test_indices], dataset[test_indices]

training_data_LOIC, testing_data_LOIC = split_dataset(data_LOIC)

print("{} examples in training, {} examples for testing.".format(
    len(training_data_LOIC), len(testing_data_LOIC)))
```

```
5565192 examples in training, 2383556 examples for testing.
```

[45]:

```
# Konvertieren des Panda Dataframes in ein TensorFlow Dataset
print("Converting Panda Dataframe into TensorFlow Dataset...")
training_dataset_LOIC = tfdf.keras.
 ↪pd_dataframe_to_tf_dataset(training_data_LOIC, label=label)
testing_dataset_LOIC = tfdf.keras.pd_dataframe_to_tf_dataset(testing_data_LOIC,␣
 ↪label=label)
```

```
Converting Panda Dataframe into TensorFlow Dataset…
```

[46]:

```
# Erstellen des Random Forest Modells
model = tfdf.keras.RandomForestModel()
model.compile(metrics=["accuracy"])
```

[47]:

```
# Trainieren des Modells
print("Training the Model...")
with sys_pipes():
    model.fit(x=training_dataset_LOIC)
```

```
Training the Model…
86957/86957 [==============================] - 73s 833us/step
[INFO kernel.cc:746] Start Yggdrasil model training
[INFO kernel.cc:747] Collect training examples
[INFO kernel.cc:392] Number of batches: 86957
[INFO kernel.cc:393] Number of examples: 5565192
[INFO kernel.cc:769] Dataset:
Number of records: 5565192
Number of columns: 5

Number of columns by type:
        NUMERICAL: 4 (80%)
        CATEGORICAL: 1 (20%)

Columns:

NUMERICAL: 4 (80%)
        0: "bwd_pkt_len_std" NUMERICAL mean:142.4 min:0 max:20469.6 sd:217.746
        1: "flow_duration" NUMERICAL mean:1.35374e+07 min:0 max:1.2e+08
sd:3.24438e+07
```

```
        2: "flow_iat_std" NUMERICAL mean:1.13545e+06 min:0 max:8.48081e+07
sd:3.80227e+06
        3: "pkt_size_avg" NUMERICAL mean:94.1061 min:0 max:8225.8 sd:106.205

CATEGORICAL: 1 (20%)
        4: "__LABEL" CATEGORICAL integerized vocab-size:3 no-ood-item

Terminology:
        nas: Number of non-available (i.e. missing) values.
        ood: Out of dictionary.
        manually-defined: Attribute which type is manually defined by the user
i.e. the type was not automatically inferred.
        tokenized: The attribute value is obtained through tokenization.
        has-dict: The attribute is attached to a string dictionary e.g. a
categorical attribute stored as a string.
        vocab-size: Number of unique values.

[INFO kernel.cc:772] Configure learner
[INFO kernel.cc:797] Training config:
learner: "RANDOM_FOREST"
features: "bwd_pkt_len_std"
features: "flow_duration"
features: "flow_iat_std"
features: "pkt_size_avg"
label: "__LABEL"
task: CLASSIFICATION
[yggdrasil_decision_forests.model.random_forest.proto.random_forest_config] {
  num_trees: 300
  decision_tree {
    max_depth: 16
    min_examples: 5
    in_split_min_examples_check: true
    missing_value_policy: GLOBAL_IMPUTATION
    allow_na_conditions: false
    categorical_set_greedy_forward {
      sampling: 0.1
      max_num_items: -1
      min_item_frequency: 1
    }
    growing_strategy_local {
    }
    categorical {
      cart {
      }
    }
    num_candidate_attributes_ratio: -1
    axis_aligned_split {
    }
    internal {
      sorting_strategy: PRESORTED
    }
  }
  winner_take_all_inference: true
  compute_oob_performances: true
  compute_oob_variable_importances: false
  adapt_bootstrap_size_ratio_for_maximum_training_duration: false
}

[INFO kernel.cc:800] Deployment config:

[INFO kernel.cc:837] Train model
[INFO random_forest.cc:303] Training random forest on 5565192 example(s) and 4
feature(s).
[INFO random_forest.cc:578] Training of tree  1/300 (tree index:2) done
accuracy:0.996984 logloss:0.108703
[INFO random_forest.cc:578] Training of tree  11/300 (tree index:9) done
accuracy:0.997088 logloss:0.0898228
[INFO random_forest.cc:578] Training of tree  19/300 (tree index:19) done
accuracy:0.997105 logloss:0.0851756
[INFO random_forest.cc:578] Training of tree  29/300 (tree index:29) done
accuracy:0.997113 logloss:0.0812797
[INFO random_forest.cc:578] Training of tree  37/300 (tree index:37) done
accuracy:0.997113 logloss:0.0795389
[INFO random_forest.cc:578] Training of tree  47/300 (tree index:46) done
accuracy:0.997118 logloss:0.0778509
[INFO random_forest.cc:578] Training of tree  55/300 (tree index:54) done
accuracy:0.997113 logloss:0.0767058
[INFO random_forest.cc:578] Training of tree  65/300 (tree index:64) done
accuracy:0.997115 logloss:0.0752676
[INFO random_forest.cc:578] Training of tree  73/300 (tree index:72) done
accuracy:0.997115 logloss:0.0745434
[INFO random_forest.cc:578] Training of tree  83/300 (tree index:82) done
accuracy:0.997116 logloss:0.0736428
[INFO random_forest.cc:578] Training of tree  91/300 (tree index:90) done
accuracy:0.997112 logloss:0.0729741
[INFO random_forest.cc:578] Training of tree  99/300 (tree index:99) done
accuracy:0.99711 logloss:0.0723139
[INFO random_forest.cc:578] Training of tree  107/300 (tree index:106) done
accuracy:0.99711 logloss:0.0718316
[INFO random_forest.cc:578] Training of tree  116/300 (tree index:115) done
accuracy:0.997113 logloss:0.0709375
[INFO random_forest.cc:578] Training of tree  124/300 (tree index:123) done
accuracy:0.99711 logloss:0.07053
[INFO random_forest.cc:578] Training of tree  133/300 (tree index:130) done
accuracy:0.99711 logloss:0.0699864
[INFO random_forest.cc:578] Training of tree  141/300 (tree index:140) done
```

```
accuracy:0.997108 logloss:0.0694699
[INFO random_forest.cc:578] Training of tree  148/300 (tree index:147) done
accuracy:0.997109 logloss:0.0691229
[INFO random_forest.cc:578] Training of tree  157/300 (tree index:156) done
accuracy:0.997107 logloss:0.0687518
[INFO random_forest.cc:578] Training of tree  166/300 (tree index:165) done
accuracy:0.997107 logloss:0.0683366
[INFO random_forest.cc:578] Training of tree  175/300 (tree index:174) done
accuracy:0.997109 logloss:0.0680178
[INFO random_forest.cc:578] Training of tree  184/300 (tree index:184) done
accuracy:0.997109 logloss:0.0676068
[INFO random_forest.cc:578] Training of tree  194/300 (tree index:192) done
accuracy:0.997107 logloss:0.0671352
[INFO random_forest.cc:578] Training of tree  201/300 (tree index:200) done
accuracy:0.997108 logloss:0.0668059
[INFO random_forest.cc:578] Training of tree  208/300 (tree index:208) done
accuracy:0.99711 logloss:0.0665099
[INFO random_forest.cc:578] Training of tree  218/300 (tree index:217) done
accuracy:0.997107 logloss:0.06618
[INFO random_forest.cc:578] Training of tree  225/300 (tree index:224) done
accuracy:0.997107 logloss:0.0659819
[INFO random_forest.cc:578] Training of tree  234/300 (tree index:232) done
accuracy:0.997107 logloss:0.0656627
[INFO random_forest.cc:578] Training of tree  243/300 (tree index:242) done
accuracy:0.997111 logloss:0.0654126
[INFO random_forest.cc:578] Training of tree  252/300 (tree index:253) done
accuracy:0.997112 logloss:0.0651231
[INFO random_forest.cc:578] Training of tree  261/300 (tree index:260) done
accuracy:0.997112 logloss:0.0648948
[INFO random_forest.cc:578] Training of tree  270/300 (tree index:269) done
accuracy:0.997111 logloss:0.0647267
[INFO random_forest.cc:578] Training of tree  278/300 (tree index:278) done
accuracy:0.997111 logloss:0.0645637
[INFO random_forest.cc:578] Training of tree  288/300 (tree index:287) done
accuracy:0.997113 logloss:0.0642322
[INFO random_forest.cc:578] Training of tree  295/300 (tree index:294) done
accuracy:0.997111 logloss:0.064033
[INFO random_forest.cc:578] Training of tree  300/300 (tree index:299) done
accuracy:0.997112 logloss:0.0639348
[INFO random_forest.cc:645] Final OOB metrics: accuracy:0.997112
logloss:0.0639348
[INFO kernel.cc:856] Export model in log directory: /tmp/tmpraei2wh4
[INFO kernel.cc:864] Save model in resources
[INFO kernel.cc:960] Loading model from path
[INFO decision_forest.cc:590] Model loaded with 300 root(s), 586268 node(s), and
4 input feature(s).
[INFO abstract_model.cc:973] Engine "RandomForestOptPred" built
[INFO kernel.cc:820] Use fast generic engine
```

[48]:
```python
# Evaluieren des trainierten Modells
print("Evaluating the Model...")
evaluation = model.evaluate(testing_dataset_LOIC, return_dict=True)
print()
for name, value in evaluation.items():
    print(f"{name}: {value:.4f}")
```

```
Evaluating the Model…
37244/37244 [==============================] - 90s 2ms/step - loss: 0.0000e+00 -
accuracy: 0.9971

loss: 0.0000
accuracy: 0.9971
```

[49]:
```python
data_path = "../Data"
model_path = "Models"
onnx_path = "ONNX_Models"
model_name = "loic_model"

# Trainiertes Modell für die spätere Verwendung abspeichern
model.save(os.path.join(data_path,model_path,model_name),overwrite=True)

# Konvertieren in das ONNX Modell
# onnx_model = k2o.convert_keras(model,df_model_name)
# onnx.save_model(onnx_model,os.path.join(data_path,onnx_path,model_name + ".
 ↪onnx"))
```

```
INFO:tensorflow:Assets written to: ../Data/Models/loic_model/assets
INFO:tensorflow:Assets written to: ../Data/Models/loic_model/assets
```

[50]:
```python
# Plotten des ersten Baumes innerhalb des Decision Forests
with open('../Data/Models/LOIC_Model_Tree.html', 'w') as f:
    f.write(tfdf.model_plotter.plot_model(model, tree_idx=0, max_depth=3))
tfdf.model_plotter.plot_model(model, tree_idx=0, max_depth=3)
```

[50]:
```
'\n<script src="https://d3js.org/d3.v6.min.js"></script>\n<div
id="tree_plot_ad86c3d8b8ba497b9498b329f4bb4379"></div>\n<script>\n/*\n *
Copyright 2021 Google LLC.\n * Licensed under the Apache License, Version 2.0
(the "License");\n * you may not use this file except in compliance with the
License.\n * You may obtain a copy of the License at\n *\n *
https://www.apache.org/licenses/LICENSE-2.0\n *\n * Unless required by
applicable law or agreed to in writing, software\n * distributed under the
License is distributed on an "AS IS" BASIS,\n * WITHOUT WARRANTIES OR CONDITIONS
OF ANY KIND, either express or implied.\n * See the License for the specific
language governing permissions and\n * limitations under the License.\n
*/\n\n/**\n *  Plotting of decision trees generated by TF-DF.\n *\n *  A tree is
a recursive structure of node objects.\n *  A node contains one or more of the
```

```
following components:\n *\n *    - A value: Representing the output of the node.
If the node is not a leaf,\n *      the value is only present for analysis i.e.
it is not used for\n *      predictions.\n *\n *    - A condition : For non-leaf
nodes, the condition (also known as split)\n *      defines a binary test to
branch to the positive or negative child.\n *\n *    - An explanation: Generally
a plot showing the relation between the label\n *      and the condition to give
insights about the effect of the condition.\n *\n *    - Two children : For non-
leaf nodes, the children nodes. The first\n *      children (i.e.
"node.children[0]") is the negative children (drawn in\n *      red). The second
children is the positive one (drawn in green).\n *\n */\n\n/**\n * Plots a
single decision tree into a DOM element.\n * @param {!options} options
Dictionary of configurations.\n * @param {!tree} raw_tree Recursive tree
structure.\n * @param {string} canvas_id Id of the output dom element.\n
*/\nfunction display_tree(options, raw_tree, canvas_id) {\n
console.log(options);\n\n  // Determine the node placement.\n  const tree_struct
= d3.tree().nodeSize(\n      [options.node_y_offset,
options.node_x_offset])(d3.hierarchy(raw_tree));\n\n  // Boundaries of the node
placement.\n  let x_min = Infinity;\n  let x_max = -x_min;\n  let y_min =
Infinity;\n  let y_max = -x_min;\n\n  tree_struct.each(d => {\n    if (d.x >
x_max) x_max = d.x;\n    if (d.x < x_min) x_min = d.x;\n    if (d.y > y_max)
y_max = d.y;\n    if (d.y < y_min) y_min = d.y;\n  });\n\n  // Size of the
plot.\n  const width = y_max - y_min + options.node_x_size + options.margin *
2;\n  const height = x_max - x_min + options.node_y_size + options.margin * 2
+\n      options.node_y_offset - options.node_y_size;\n\n  const plot =
d3.select(canvas_id);\n\n  // Tool tip\n  options.tooltip =
plot.append(\'div\')\n                       .attr(\'width\', 100)\n
.attr(\'height\', 100)\n                       .style(\'padding\', \'4px\')\n
.style(\'background\', \'#fff\')\n                       .style(\'box-shadow\',
\'4px 4px 0px rgba(0,0,0,0.1)\')\n                       .style(\'border\',
\'1px solid black\')\n                       .style(\'font-family\', \'sans-
serif\')\n                       .style(\'font-size\', options.font_size)\n
.style(\'position\', \'absolute\')\n                       .style(\'z-index\',
\'10\')\n                       .attr(\'pointer-events\', \'none\')\n
.style(\'display\', \'none\');\n\n  // Create canvas\n  const svg =
plot.append(\'svg\').attr(\'width\', width).attr(\'height\', height);\n  const
graph =\n      svg.style(\'overflow\', \'visible\')\n          .append(\'g\')\n
.attr(\'font-family\', \'sans-serif\')\n          .attr(\'font-size\',
options.font_size)\n          .attr(\n              \'transform\',\n
() => `translate(${options.margin},${\n                  - x_min +
options.node_y_offset / 2 + options.margin})`);\n\n  // Plot bounding box.\n  if
(options.show_plot_bounding_box) {\n    svg.append(\'rect\')\n
.attr(\'width\', width)\n        .attr(\'height\', height)\n
.attr(\'fill\', \'none\')\n        .attr(\'stroke-width\', 1.0)\n
.attr(\'stroke\', \'black\');\n  }\n\n  // Draw the edges.\n
display_edges(options, graph, tree_struct);\n\n  // Draw the nodes.\n
display_nodes(options, graph, tree_struct);\n}\n\n/**\n * Draw the nodes of the
tree.\n * @param {!options} options Dictionary of configurations.\n * @param
```

```
{!graph} graph D3 search handle containing the graph.\n * @param {!tree_struct}
tree_struct Structure of the tree (node placement,\n *     data, etc.).\n
*/\nfunction display_nodes(options, graph, tree_struct) {\n  const nodes =
graph.append(\'g\')\n                    .selectAll(\'g\')\n
.data(tree_struct.descendants())\n                    .join(\'g\')\n
.attr(\'transform\', d => `translate(${d.y},${d.x})`);\n\n
nodes.append(\'rect\')\n      .attr(\'x\', 0.5)\n      .attr(\'y\', 0.5)\n
.attr(\'width\', options.node_x_size)\n      .attr(\'height\',
options.node_y_size)\n      .attr(\'stroke\', \'lightgrey\')\n
.attr(\'stroke-width\', 1)\n      .attr(\'fill\', \'white\')\n      .attr(\'y\',
-options.node_y_size / 2);\n\n  // Brackets on the right of condition nodes
without children.\n  non_leaf_node_without_children =\n      nodes.filter(node
=> node.data.condition != null && node.children == null)\n
.append(\'g\')\n          .attr(\'transform\',
`translate(${options.node_x_size},0)`);\n\n
non_leaf_node_without_children.append(\'path\')\n      .attr(\'d\', \'M0,0 C
10,0 0,10 10,10\')\n      .attr(\'fill\', \'none\')\n      .attr(\'stroke-
width\', 1.0)\n      .attr(\'stroke\', \'#F00\');\n\n
non_leaf_node_without_children.append(\'path\')\n      .attr(\'d\', \'M0,0 C
10,0 0,-10 10,-10\')\n      .attr(\'fill\', \'none\')\n      .attr(\'stroke-
width\', 1.0)\n      .attr(\'stroke\', \'#0F0\');\n\n  const node_content =
nodes.append(\'g\').attr(\n      \'transform\',\n
`translate(0,${options.node_padding - options.node_y_size / 2})`);\n\n
node_content.append(node => create_node_element(options, node));\n}\n\n/**\n *
Creates the D3 content for a single node.\n * @param {!options} options
Dictionary of configurations.\n * @param {!node} node Node to draw.\n * @return
{!d3} D3 content.\n */\nfunction create_node_element(options, node) {\n  //
Output accumulator.\n  let output = {\n    // Content to draw.\n    content:
d3.create(\'svg:g\'),\n    // Vertical offset to the next element to draw.\n
vertical_offset: 0\n  };\n\n  // Conditions.\n  if (node.data.condition != null)
{\n    display_condition(options, node.data.condition, output);\n  }\n\n  //
Values.\n  if (node.data.value != null) {\n    display_value(options,
node.data.value, output);\n  }\n\n  // Explanations.\n  if
(node.data.explanation != null) {\n    display_explanation(options,
node.data.explanation, output);\n  }\n\n  return
output.content.node();\n}\n\n\n/**\n * Adds a single line of text inside of a
node.\n * @param {!options} options Dictionary of configurations.\n * @param
{string} text Text to display.\n * @param {!output} output Output display
accumulator.\n */\nfunction display_node_text(options, text, output) {\n
output.content.append(\'text\')\n      .attr(\'x\', options.node_padding)\n
.attr(\'y\', output.vertical_offset)\n      .attr(\'alignment-baseline\',
\'hanging\')\n      .text(text);\n  output.vertical_offset += 10;\n}\n\n/**\n *
Adds a single line of text inside of a node with a tooltip.\n * @param
{!options} options Dictionary of configurations.\n * @param {string} text Text
to display.\n * @param {string} tooltip Text in the Tooltip.\n * @param
{!output} output Output display accumulator.\n */\nfunction
display_node_text_with_tooltip(options, text, tooltip, output) {\n  const item =
output.content.append(\'text\')\n                  .attr(\'x\',
options.node_padding)\n                  .attr(\'alignment-baseline\',
\'hanging\')\n                  .text(text);\n\n  add_tooltip(options, item, ()
=> tooltip);\n  output.vertical_offset += 10;\n}\n\n/**\n * Adds a tooltip to a
dom element.\n * @param {!options} options Dictionary of configurations.\n *
@param {!dom} target Dom element to equip with a tooltip.\n * @param {!func}
get_content Generates the html content of the tooltip.\n */\nfunction
add_tooltip(options, target, get_content) {\n  function show(d) {\n
options.tooltip.style(\'display\', \'block\');\n
options.tooltip.html(get_content());\n  }\n\n  function hide(d) {\n
options.tooltip.style(\'display\', \'none\');\n  }\n\n  function move(d) {\n
options.tooltip.style(\'display\', \'block\');\n
options.tooltip.style(\'left\', (d.pageX + 5) + \'px\');\n
options.tooltip.style(\'top\', d.pageY + \'px\');\n  }\n\n
target.on(\'mouseover\', show);\n  target.on(\'mouseout\', hide);\n
target.on(\'mousemove\', move);\n}\n\n/**\n * Adds a condition inside of a
node.\n * @param {!options} options Dictionary of configurations.\n * @param
{!condition} condition Condition to display.\n * @param {!output} output Output
display accumulator.\n */\nfunction display_condition(options, condition,
output) {\n  threshold_format = d3.format(\'r\');\n\n  if (condition.type ===
\'IS_MISSING\') {\n    display_node_text(options, `${condition.attribute} is
missing`, output);\n    return;\n  }\n\n  if (condition.type === \'IS_TRUE\')
{\n    display_node_text(options, `${condition.attribute} is true`, output);\n
return;\n  }\n\n  if (condition.type === \'NUMERICAL_IS_HIGHER_THAN\') {\n
format = d3.format(\'r\');\n    display_node_text(\n      options,\n
`${condition.attribute} >= ${threshold_format(condition.threshold)}`,\n
output);\n    return;\n  }\n\n  if (condition.type === \'CATEGORICAL_IS_IN\')
{\n    display_node_text_with_tooltip(\n      options, `${condition.attribute}
in […]`,\n      `${condition.attribute} in [${condition.mask}]`, output);\n
return;\n  }\n\n  if (condition.type === \'CATEGORICAL_SET_CONTAINS\') {\n
display_node_text_with_tooltip(\n      options, `${condition.attribute}
intersect […]`,\n      `${condition.attribute} intersect
[${condition.mask}]`, output);\n    return;\n  }\n\n  if (condition.type ===
\'NUMERICAL_SPARSE_OBLIQUE\') {\n    display_node_text_with_tooltip(\n
options, `Sparse oblique split…`,\n
`[${condition.attributes}]*[${condition.weights}]>=${\n
threshold_format(condition.threshold)}`,\n      output);\n    return;\n  }\n\n
display_node_text(\n     options, `Non supported condition ${condition.type}`,
output);\n}\n\n/**\n * Adds a value inside of a node.\n * @param {!options}
options Dictionary of configurations.\n * @param {!value} value Value to
display.\n * @param {!output} output Output display accumulator.\n */\nfunction
display_value(options, value, output) {\n  if (value.type === \'PROBABILITY\')
{\n    const left_margin = 0;\n    const right_margin = 50;\n    const
plot_width = options.node_x_size - options.node_padding * 2 -\n
left_margin - right_margin;\n\n    let cusum =
Array.from(d3.cumsum(value.distribution));\n    cusum.unshift(0);\n    const
distribution_plot = output.content.append(\'g\').attr(\n      \'transform\',
`translate(0,${output.vertical_offset + 0.5})`);\n\n
distribution_plot.selectAll(\'rect\')\n        .data(value.distribution)\n
.join(\'rect\')\n        .attr(\'height\', 10)\n        .attr(\n
\'x\',\n            (d, i) =>\n                (cusum[i] * plot_width +
left_margin + options.node_padding))\n        .attr(\'width\', (d, i) => d *
plot_width)\n        .style(\'fill\', (d, i) => d3.schemeSet1[i]);\n\n    const
num_examples =\n        output.content.append(\'g\')\n
.attr(\'transform\', `translate(0,${output.vertical_offset})`)\n
.append(\'text\')\n            .attr(\'x\', options.node_x_size -
options.node_padding)\n            .attr(\'alignment-baseline\', \'hanging\')\n
.attr(\'text-anchor\', \'end\')\n
.text(`(${value.num_examples})`);\n\n    const distribution_details =
d3.create(\'ul\');\n    distribution_details.selectAll(\'li\')\n
.data(value.distribution)\n        .join(\'li\')\n        .append(\'span\')\n
.text(\n            (d, i) =>\n                \'class \' + i + \': \' +
d3.format(\'.3%\')(value.distribution[i]));\n\n    add_tooltip(options,
distribution_plot, () => distribution_details.html());\n    add_tooltip(options,
num_examples, () => \'Number of examples\');\n\n    output.vertical_offset +=
10;\n    return;\n  }\n\n  if (value.type === \'REGRESSION\') {\n
display_node_text(\n        options,\n        \'value: \' +
d3.format(\'r\')(value.value) + ` (` +\n
d3.format(\'.6\')(value.num_examples) + `)`,\n        output);\n    return;\n
}\n\n  display_node_text(options, `Non supported value ${value.type}`,
output);\n}\n\n/**\n * Adds an explanation inside of a node.\n * @param
{!options} options Dictionary of configurations.\n * @param {!explanation}
explanation Explanation to display.\n * @param {!output} output Output display
accumulator.\n */\nfunction display_explanation(options, explanation, output)
{\n  // Margin before the explanation.\n  output.vertical_offset += 10;\n\n
display_node_text(\n      options, `Non supported explanation
${explanation.type}`, output);\n}\n\n\n/**\n * Draw the edges of the tree.\n *
@param {!options} options Dictionary of configurations.\n * @param {!graph}
graph D3 search handle containing the graph.\n * @param {!tree_struct}
tree_struct Structure of the tree (node placement,\n *     data, etc.).\n
*/\nfunction display_edges(options, graph, tree_struct) {\n  // Draw an edge
between a parent and a child node with a bezier.\n  function draw_single_edge(d)
{\n    return \'M\' + (d.source.y + options.node_x_size) + \',\' + d.source.x +
\' C\' +\n        (d.source.y + options.node_x_size + options.edge_rounding) +
\',\' +\n        d.source.x + \' \' + (d.target.y - options.edge_rounding) +
\',\' +\n        d.target.x + \' \' + d.target.y + \',\' + d.target.x;\n  }\n\n
graph.append(\'g\')\n      .attr(\'fill\', \'none\')\n      .attr(\'stroke-
width\', 1.2)\n      .selectAll(\'path\')\n      .data(tree_struct.links())\n
.join(\'path\')\n      .attr(\'d\', draw_single_edge)\n      .attr(\n
\'stroke\', d => (d.target === d.source.children[0]) ? \'#0F0\' :
\'#F00\');\n}\n\ndisplay_tree({"margin": 10, "node_x_size": 160, "node_y_size":
28, "node_x_offset": 180, "node_y_offset": 33, "font_size": 10, "edge_rounding":
20, "node_padding": 2, "show_plot_bounding_box": false}, {"value": {"type":
"PROBABILITY", "distribution": [0.927539067834497, 0.072460932165503],
"num_examples": 5565192.0}, "condition": {"type": "NUMERICAL_IS_HIGHER_THAN",
"attribute": "bwd_pkt_len_std", "threshold": 481.995361328125}, "children":
[{"value": {"type": "PROBABILITY", "distribution": [0.6928326510381012,
0.30716734896189873], "num_examples": 661737.0}, "condition": {"type":
"NUMERICAL_IS_HIGHER_THAN", "attribute": "bwd_pkt_len_std", "threshold":
482.0030212402344}, "children": [{"value": {"type": "PROBABILITY",
"distribution": [1.0, 0.0], "num_examples": 458470.0}}, {"value": {"type":
"PROBABILITY", "distribution": [1.4758913153635367e-05, 0.9999852410868464],
"num_examples": 203267.0}, "condition": {"type": "NUMERICAL_IS_HIGHER_THAN",
"attribute": "flow_iat_std", "threshold": 798255.1875}, "children": [{"value":
{"type": "PROBABILITY", "distribution": [0.6, 0.4], "num_examples": 5.0}},
{"value": {"type": "PROBABILITY", "distribution": [0.0, 1.0], "num_examples":
203262.0}}]}]}, {"value": {"type": "PROBABILITY", "distribution":
[0.959213452555392, 0.04078654744460793], "num_examples": 4903455.0},
"condition": {"type": "NUMERICAL_IS_HIGHER_THAN", "attribute": "flow_duration",
"threshold": 6012082.5}, "children": [{"value": {"type": "PROBABILITY",
"distribution": [0.7797676589112654, 0.22023234108873457], "num_examples":
778941.0}, "condition": {"type": "NUMERICAL_IS_HIGHER_THAN", "attribute":
"flow_iat_std", "threshold": 2.101493120193481 4}, "children": [{"value":
{"type": "PROBABILITY", "distribution": [0.998816961210913 1,
0.0011830387890868724], "num_examples": 493644.0}, "condition": {"type":
"NUMERICAL_IS_HIGHER_THAN", "attribute": "flow_iat_std", "threshold":
66992.890625}}, {"value": {"type": "PROBABILITY", "distribution":
[0.4007507965383443, 0.5992492034616558], "num_examples": 285297.0},
"condition": {"type": "NUMERICAL_IS_HIGHER_THAN", "attribute": "flow_duration",
"threshold": 50766992.0}}]}, {"value": {"type": "PROBABILITY", "distribution":
[0.9931029449772749, 0.006897055022725101], "num_examples": 4124514.0},
"condition": {"type": "NUMERICAL_IS_HIGHER_THAN", "attribute": "pkt_size_avg",
"threshold": 0.6614583730697632}, "children": [{"value": {"type": "PROBABILITY",
"distribution": [0.9999977725981847, 2.2274018153324795e-06], "num_examples":
3142675.0}, "condition": {"type": "NUMERICAL_IS_HIGHER_THAN", "attribute":
"flow_duration", "threshold": 456440.5}}, {"value": {"type": "PROBABILITY",
"distribution": [0.9710339475209275, 0.028966052479072434], "num_examples":
981839.0}, "condition": {"type": "NUMERICAL_IS_HIGHER_THAN", "attribute":
"flow_duration", "threshold": 371752.5}}]}]}]},
"#tree_plot_ad86c3d8b8ba497b9498b329f4bb4379")\n</script>\n'
```

[51]:
```
# Erstellen einer Bilanz für das trainierte Modell
model.summary()
```

```
Model: "random_forest_model_2"

_________________________________________________________________
Layer (type)                Output Shape              Param #
=================================================================
Total params: 1
Trainable params: 0
Non-trainable params: 1
```

```
-----------------------------------------------------------------
Type: "RANDOM_FOREST"
Task: CLASSIFICATION
Label: "__LABEL"

Input Features (4):
        bwd_pkt_len_std
        flow_duration
        flow_iat_std
        pkt_size_avg

No weights

Variable Importance: NUM_NODES:
    1.   "flow_duration" 285691.000000 ################
    2.    "pkt_size_avg" 3430.000000
    3.    "flow_iat_std" 2718.000000
    4. "bwd_pkt_len_std" 1145.000000

Variable Importance: NUM_AS_ROOT:
    1.   "flow_duration" 234.000000 ################
    2. "bwd_pkt_len_std" 66.000000

Variable Importance: SUM_SCORE:
    1.   "flow_duration" 159002490.749388 ################
    2.    "pkt_size_avg" 137233041.642846 ############
    3.    "flow_iat_std" 69001430.202241 ##
    4. "bwd_pkt_len_std" 55517544.016287

Variable Importance: MEAN_MIN_DEPTH:
    1.         "__LABEL" 14.037804 ################
    2. "bwd_pkt_len_std"  7.742834 ########
    3.    "flow_iat_std"  3.205292 ###
    4.    "pkt_size_avg"  2.031479 #
    5.   "flow_duration"  0.390593



Winner take all: true
Out-of-bag evaluation: accuracy:0.997112 logloss:0.0639348
Number of trees: 300
Total number of nodes: 586268

Number of nodes by tree:
Count: 300 Average: 1954.23 StdDev: 286.042
Min: 1037 Max: 2735 Ignored: 0
----------------------------------------------
[ 1037, 1121)  1   0.33%   0.33%
[ 1121, 1206)  1   0.33%   0.67%
[ 1206, 1291)  1   0.33%   1.00%
[ 1291, 1376)  5   1.67%   2.67% #
[ 1376, 1461)  9   3.00%   5.67% ##
[ 1461, 1546) 14   4.67%  10.33% ###
[ 1546, 1631) 10   3.33%  13.67% ##
[ 1631, 1716) 18   6.00%  19.67% ####
[ 1716, 1801) 26   8.67%  28.33% #####
[ 1801, 1886) 26   8.67%  37.00% #####
[ 1886, 1971) 29   9.67%  46.67% ######
[ 1971, 2056) 39  13.00%  59.67% ########
[ 2056, 2141) 50  16.67%  76.33% ##########
[ 2141, 2226) 32  10.67%  87.00% ######
[ 2226, 2311) 20   6.67%  93.67% ####
[ 2311, 2396)  6   2.00%  95.67% #
[ 2396, 2481)  1   0.33%  96.00%
[ 2481, 2566)  3   1.00%  97.00% #
[ 2566, 2651)  4   1.33%  98.33% #
[ 2651, 2735]  5   1.67% 100.00% #

Depth by leafs:
Count: 293284 Average: 14.0391 StdDev: 1.83405
Min: 2 Max: 15 Ignored: 0
----------------------------------------------
[  2,  3)    271   0.09%   0.09%
[  3,  4)    448   0.15%   0.25%
[  4,  5)   1070   0.36%   0.61%
[  5,  6)   1370   0.47%   1.08%
[  6,  7)   1320   0.45%   1.53%
[  7,  8)   1624   0.55%   2.08%
[  8,  9)   1468   0.50%   2.58%
[  9, 10)   2325   0.79%   3.37%
[ 10, 11)   3702   1.26%   4.64%
[ 11, 12)   7692   2.62%   7.26%
[ 12, 13)  15549   5.30%  12.56% #
[ 13, 14)  28034   9.56%  22.12% ##
[ 14, 15)  46357  15.81%  37.93% ###
[ 15, 15] 182054  62.07% 100.00% ##########

Number of training obs by leaf:
Count: 293284 Average: 5692.63 StdDev: 89597
Min: 5 Max: 3303393 Ignored: 0
----------------------------------------------
[       5,  165174) 291677  99.45%  99.45% ##########
[  165174,  330343)    435   0.15%  99.60%
[  330343,  495513)    178   0.06%  99.66%
[  495513,  660682)    198   0.07%  99.73%
[  660682,  825852)    311   0.11%  99.83%
[  825852,  991021)      8   0.00%  99.84%
[  991021, 1156191)    177   0.06%  99.90%
[ 1156191, 1321360)      0   0.00%  99.90%
[ 1321360, 1486530)      0   0.00%  99.90%
[ 1486530, 1651699)      4   0.00%  99.90%
[ 1651699, 1816868)      2   0.00%  99.90%
[ 1816868, 1982038)      0   0.00%  99.90%
[ 1982038, 2147207)      2   0.00%  99.90%
[ 2147207, 2312377)    145   0.05%  99.95%
[ 2312377, 2477546)     88   0.03%  99.98%
[ 2477546, 2642716)      2   0.00%  99.98%
[ 2642716, 2807885)      1   0.00%  99.98%
[ 2807885, 2973055)     15   0.01%  99.99%
[ 2973055, 3138224)      2   0.00%  99.99%
[ 3138224, 3303393]     39   0.01% 100.00%

Attribute in nodes:
        285691 : flow_duration [NUMERICAL]
        3430 : pkt_size_avg [NUMERICAL]
        2718 : flow_iat_std [NUMERICAL]
        1145 : bwd_pkt_len_std [NUMERICAL]

Attribute in nodes with depth <= 0:
        234 : flow_duration [NUMERICAL]
        66 : bwd_pkt_len_std [NUMERICAL]

Attribute in nodes with depth <= 1:
        416 : flow_duration [NUMERICAL]
        246 : pkt_size_avg [NUMERICAL]
        181 : bwd_pkt_len_std [NUMERICAL]
        57 : flow_iat_std [NUMERICAL]

Attribute in nodes with depth <= 2:
        643 : pkt_size_avg [NUMERICAL]
        632 : flow_duration [NUMERICAL]
        439 : bwd_pkt_len_std [NUMERICAL]
        115 : flow_iat_std [NUMERICAL]

Attribute in nodes with depth <= 3:
        998 : pkt_size_avg [NUMERICAL]
        904 : flow_duration [NUMERICAL]
        763 : bwd_pkt_len_std [NUMERICAL]
        574 : flow_iat_std [NUMERICAL]

Attribute in nodes with depth <= 5:
        2760 : flow_duration [NUMERICAL]
        1642 : pkt_size_avg [NUMERICAL]
        1640 : flow_iat_std [NUMERICAL]
        1077 : bwd_pkt_len_std [NUMERICAL]

Condition type in nodes:
        292984 : HigherCondition
Condition type in nodes with depth <= 0:
        300 : HigherCondition
Condition type in nodes with depth <= 1:
        900 : HigherCondition
Condition type in nodes with depth <= 2:
        1829 : HigherCondition
Condition type in nodes with depth <= 3:
        3239 : HigherCondition
Condition type in nodes with depth <= 5:
        7119 : HigherCondition
Node format: NOT_SET

Training OOB:
        trees: 1, Out-of-bag evaluation: accuracy:0.996984 logloss:0.108703
        trees: 11, Out-of-bag evaluation: accuracy:0.997088 logloss:0.0898228
        trees: 19, Out-of-bag evaluation: accuracy:0.997105 logloss:0.0851756
        trees: 29, Out-of-bag evaluation: accuracy:0.997113 logloss:0.0812797
        trees: 37, Out-of-bag evaluation: accuracy:0.997113 logloss:0.0795389
        trees: 47, Out-of-bag evaluation: accuracy:0.997118 logloss:0.0778509
        trees: 55, Out-of-bag evaluation: accuracy:0.997113 logloss:0.0767058
        trees: 65, Out-of-bag evaluation: accuracy:0.997115 logloss:0.0752676
        trees: 73, Out-of-bag evaluation: accuracy:0.997115 logloss:0.0745434
        trees: 83, Out-of-bag evaluation: accuracy:0.997116 logloss:0.0736428
        trees: 91, Out-of-bag evaluation: accuracy:0.997112 logloss:0.0729741
        trees: 99, Out-of-bag evaluation: accuracy:0.99711 logloss:0.0723139
        trees: 107, Out-of-bag evaluation: accuracy:0.99711 logloss:0.0718316
        trees: 116, Out-of-bag evaluation: accuracy:0.997113 logloss:0.0709375
        trees: 124, Out-of-bag evaluation: accuracy:0.99711 logloss:0.07053
        trees: 133, Out-of-bag evaluation: accuracy:0.99711 logloss:0.0699864
        trees: 141, Out-of-bag evaluation: accuracy:0.997108 logloss:0.0694699
        trees: 148, Out-of-bag evaluation: accuracy:0.997109 logloss:0.0691229
        trees: 157, Out-of-bag evaluation: accuracy:0.997107 logloss:0.0687518
        trees: 166, Out-of-bag evaluation: accuracy:0.997107 logloss:0.0683366
        trees: 175, Out-of-bag evaluation: accuracy:0.997109 logloss:0.0680178
        trees: 184, Out-of-bag evaluation: accuracy:0.997109 logloss:0.0676068
        trees: 194, Out-of-bag evaluation: accuracy:0.997107 logloss:0.0671352
        trees: 201, Out-of-bag evaluation: accuracy:0.997108 logloss:0.0668059
        trees: 208, Out-of-bag evaluation: accuracy:0.99711 logloss:0.0665099
        trees: 218, Out-of-bag evaluation: accuracy:0.997107 logloss:0.06618
        trees: 225, Out-of-bag evaluation: accuracy:0.997107 logloss:0.0659819
        trees: 234, Out-of-bag evaluation: accuracy:0.997107 logloss:0.0656627
        trees: 243, Out-of-bag evaluation: accuracy:0.997111 logloss:0.0654126
        trees: 252, Out-of-bag evaluation: accuracy:0.997112 logloss:0.0651231
        trees: 261, Out-of-bag evaluation: accuracy:0.997112 logloss:0.0648948
```

```
trees: 270, Out-of-bag evaluation: accuracy:0.997111 logloss:0.0647267
trees: 278, Out-of-bag evaluation: accuracy:0.997111 logloss:0.0645637
trees: 288, Out-of-bag evaluation: accuracy:0.997113 logloss:0.0642322
trees: 295, Out-of-bag evaluation: accuracy:0.997111 logloss:0.064033
trees: 300, Out-of-bag evaluation: accuracy:0.997112 logloss:0.0639348
```

[52]:

```
# Erstellen von Grafiken für die Effizienz des Trainings
logs = model.make_inspector().training_logs()
plt.figure(figsize=(12,4))

plt.subplot(1,2,1)
plt.plot([log.num_trees for log in logs], [log.evaluation.accuracy for log in␣
 ↪logs])
plt.xlabel("Number of trees")
plt.ylabel("Accuracy (out-of-bag)")

plt.subplot(1,2,2)
plt.plot([log.num_trees for log in logs], [log.evaluation.loss for log in logs])
plt.xlabel("Number of trees")
plt.ylabel("Logloss (out-of-bag)")

plt.savefig('../Data/Visualized/LOIC_Model.png')
plt.clf()
```

```
<Figure size 864x288 with 0 Axes>
```