RandomForest_Slowloris

June 29, 2021

[1]:

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

[2]:

```
# 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: 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: 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: 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: 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: 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: 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: 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: 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: 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: 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: 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: 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: 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: 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: 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: 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: 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: 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: 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: 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: 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: 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: 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: 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: 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: 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: 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<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: 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: 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: 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: 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: 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: 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: 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: 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: 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: pyasn1<0.5.0,>=0.4.6 in
/home/julianbuecher/Projects/Bachelor-
Thesis/ML.Proxy.Python.ModelTrainer/lib/python3.8/site-packages (from
pyasn1-modules>=0.2.1->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: 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: 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: 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: 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: 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: 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: 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: 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: 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: 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: flatbuffers in
/home/julianbuecher/Projects/Bachelor-
Thesis/ML.Proxy.Python.ModelTrainer/lib/python3.8/site-packages (from
onnxruntime) (1.12)
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: 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: 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: onnx in /home/julianbuecher/Projects/Bachelor-
Thesis/ML.Proxy.Python.ModelTrainer/lib/python3.8/site-packages (from
keras2onnx) (1.9.0)
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: six in /home/julianbuecher/Projects/Bachelor-
Thesis/ML.Proxy.Python.ModelTrainer/lib/python3.8/site-packages (from
onnx->keras2onnx) (1.15.0)
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)
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: 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: 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)
```

[3]:

```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
```

[4]:

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

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

[5]:

```python
# Laden der Netzwerk Traffic Daten für den GoldenEye Angriff
data_Slowloris = pd.read_csv('../Data/Thursday-15-02-2018_Slowloris-Attack.csv')
# Umbenennen der einzelnen Spalte für eine bessere Kompatibilität mit TensorFlow
data_Slowloris.rename(columns={
    'Flow Duration':'flow_duration',
    'Bwd IAT Mean':'bwd_iat_mean',
    'Fwd IAT Min':'fwd_iat_min',
    'Fwd IAT Mean':'fwd_iat_mean',
    'Label':'label'
},
inplace=True)
```

[6]:

```python
# Festlegen des Wertes der bestimmten Variable
label = 'label'
```

[7]:

```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_Slowloris, testing_data_Slowloris = split_dataset(data_Slowloris)

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

```
705517 examples in training, 301550 examples for testing.
```

[8]:

```
# Konvertieren des Panda Dataframes in ein TensorFlow Dataset
print("Converting Panda Dataframe into TensorFlow Dataset...")
training_dataset_Slowloris = tfdf.keras.
 ↪pd_dataframe_to_tf_dataset(training_data_Slowloris, label=label)
testing_dataset_Slowloris = tfdf.keras.
 ↪pd_dataframe_to_tf_dataset(testing_data_Slowloris, label=label)
```

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

[9]:

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

[10]:

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

```
Training the Model:
2021-06-29 17:07:18.429984: I
tensorflow/compiler/mlir/mlir_graph_optimization_pass.cc:176] None of the MLIR
Optimization Passes are enabled (registered 2)
2021-06-29 17:07:18.448871: I
tensorflow/core/platform/profile_utils/cpu_utils.cc:114] CPU Frequency:
2199995000 Hz
11024/11024 [==============================] - 10s 746us/step
[INFO kernel.cc:746] Start Yggdrasil model training
[INFO kernel.cc:747] Collect training examples
[INFO kernel.cc:392] Number of batches: 11024
[INFO kernel.cc:393] Number of examples: 705517
[INFO kernel.cc:769] Dataset:
Number of records: 705517
Number of columns: 5

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

```
Columns:

NUMERICAL: 4 (80%)
        0: "bwd_iat_mean" NUMERICAL mean:1.41201e+06 min:0 max:1.19913e+08
sd:7.43499e+06
        1: "flow_duration" NUMERICAL mean:1.44362e+07 min:0 max:1.2e+08
sd:3.3857e+07
        2: "fwd_iat_mean" NUMERICAL mean:3.47284e+06 min:0 max:1.19992e+08
sd:1.23694e+07
        3: "fwd_iat_min" NUMERICAL mean:2.64068e+06 min:0 max:1.19992e+08
sd:1.22671e+07

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_iat_mean"
features: "flow_duration"
features: "fwd_iat_mean"
features: "fwd_iat_min"
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 705517 example(s) and 4
feature(s).
[INFO random_forest.cc:578] Training of tree  1/300 (tree index:3) done
accuracy:0.999245 logloss:0.0272256
[INFO random_forest.cc:578] Training of tree  11/300 (tree index:8) done
accuracy:0.999411 logloss:0.0160011
[INFO random_forest.cc:578] Training of tree  21/300 (tree index:19) done
accuracy:0.999436 logloss:0.0143874
[INFO random_forest.cc:578] Training of tree  31/300 (tree index:30) done
accuracy:0.999456 logloss:0.0136232
[INFO random_forest.cc:578] Training of tree  41/300 (tree index:39) done
accuracy:0.99945 logloss:0.0128219
[INFO random_forest.cc:578] Training of tree  51/300 (tree index:49) done
accuracy:0.999447 logloss:0.0127773
[INFO random_forest.cc:578] Training of tree  61/300 (tree index:60) done
accuracy:0.999443 logloss:0.012591
[INFO random_forest.cc:578] Training of tree  71/300 (tree index:70) done
accuracy:0.999453 logloss:0.012499
[INFO random_forest.cc:578] Training of tree  81/300 (tree index:80) done
accuracy:0.999456 logloss:0.0124896
[INFO random_forest.cc:578] Training of tree  91/300 (tree index:90) done
accuracy:0.999454 logloss:0.0124424
[INFO random_forest.cc:578] Training of tree  101/300 (tree index:100) done
accuracy:0.999459 logloss:0.0123936
[INFO random_forest.cc:578] Training of tree  111/300 (tree index:108) done
accuracy:0.999457 logloss:0.0122594
[INFO random_forest.cc:578] Training of tree  121/300 (tree index:121) done
```

```
accuracy:0.999456 logloss:0.0121669
[INFO random_forest.cc:578] Training of tree  131/300 (tree index:130) done
accuracy:0.999459 logloss:0.0121653
[INFO random_forest.cc:578] Training of tree  141/300 (tree index:140) done
accuracy:0.999459 logloss:0.0120756
[INFO random_forest.cc:578] Training of tree  151/300 (tree index:152) done
accuracy:0.999459 logloss:0.0120764
[INFO random_forest.cc:578] Training of tree  161/300 (tree index:160) done
accuracy:0.999463 logloss:0.0120317
[INFO random_forest.cc:578] Training of tree  171/300 (tree index:169) done
accuracy:0.999464 logloss:0.0118936
[INFO random_forest.cc:578] Training of tree  181/300 (tree index:180) done
accuracy:0.99946 logloss:0.0118927
[INFO random_forest.cc:578] Training of tree  191/300 (tree index:190) done
accuracy:0.999461 logloss:0.0118931
[INFO random_forest.cc:578] Training of tree  201/300 (tree index:200) done
accuracy:0.999463 logloss:0.0118914
[INFO random_forest.cc:578] Training of tree  211/300 (tree index:210) done
accuracy:0.999463 logloss:0.0118918
[INFO random_forest.cc:578] Training of tree  221/300 (tree index:221) done
accuracy:0.999461 logloss:0.0118918
[INFO random_forest.cc:578] Training of tree  231/300 (tree index:230) done
accuracy:0.999463 logloss:0.0118431
[INFO random_forest.cc:578] Training of tree  241/300 (tree index:240) done
accuracy:0.999461 logloss:0.0118401
[INFO random_forest.cc:578] Training of tree  251/300 (tree index:247) done
accuracy:0.999464 logloss:0.0118391
[INFO random_forest.cc:578] Training of tree  261/300 (tree index:258) done
accuracy:0.999463 logloss:0.0118403
[INFO random_forest.cc:578] Training of tree  271/300 (tree index:271) done
accuracy:0.999466 logloss:0.011841
[INFO random_forest.cc:578] Training of tree  281/300 (tree index:280) done
accuracy:0.999467 logloss:0.0117971
[INFO random_forest.cc:578] Training of tree  291/300 (tree index:289) done
accuracy:0.999466 logloss:0.0117986
[INFO random_forest.cc:578] Training of tree  300/300 (tree index:299) done
accuracy:0.999464 logloss:0.0117982
[INFO random_forest.cc:645] Final OOB metrics: accuracy:0.999464
logloss:0.0117982
[INFO kernel.cc:856] Export model in log directory: /tmp/tmpdn_vn5u5
[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), 124818 node(s), and
4 input feature(s).
[INFO abstract_model.cc:973] Engine "RandomForestOptPred" built
[INFO kernel.cc:820] Use fast generic engine
```

[11]:

```python
# Evaluation des trainierten Modells mit den Testdaten
print("Evaluating the Model...")
evaluation = model.evaluate(testing_dataset_Slowloris, return_dict=True)

print()

for name, value in evaluation.items():
    print(f"{name}: {value:.4f}")
```

```
Evaluating the Model…
4712/4712 [==============================] - 16s 3ms/step - loss: 0.0000e+00 -
accuracy: 0.9994

loss: 0.0000
accuracy: 0.9994
```

[12]:

```python
data_path = "../Data"
model_path = "Models"
onnx_path = "ONNX_Models"
model_name = "slowloris_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/slowloris_model/assets
INFO:tensorflow:Assets written to: ../Data/Models/slowloris_model/assets
```

[13]:

```python
# Plotten des ersten Baumes innerhalb des Decision Forests
with open('../Data/Models/Slowloris_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)
```

[13]:

```
'\n<script src="https://d3js.org/d3.v6.min.js"></script>\n<div
id="tree_plot_bc16f2b0ee554b4ba184df72c01229b0"></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": [0.9890675915676022, 0.010932408432397803],
"num_examples": 705517.0}, "condition": {"type": "NUMERICAL_IS_HIGHER_THAN",
"attribute": "bwd_iat_mean", "threshold": 30995712.0}, "children": [{"value":
{"type": "PROBABILITY", "distribution": [0.337810181632131, 0.662189818367869],
"num_examples": 7818.0}, "condition": {"type": "NUMERICAL_IS_HIGHER_THAN",
"attribute": "flow_duration", "threshold": 99998064.0}, "children": [{"value":
{"type": "PROBABILITY", "distribution": [0.11475697490787858,
0.8852430250921214], "num_examples": 5699.0}, "condition": {"type":
"NUMERICAL_IS_HIGHER_THAN", "attribute": "bwd_iat_mean", "threshold":
52604816.0}, "children": [{"value": {"type": "PROBABILITY", "distribution":
[0.03573532299234243, 0.9642646770076576], "num_examples": 5093.0}, "condition":
{"type": "NUMERICAL_IS_HIGHER_THAN", "attribute": "flow_duration", "threshold":
111162960.0}}, {"value": {"type": "PROBABILITY", "distribution":
[0.7788778877887789, 0.22112211221122113], "num_examples": 606.0}, "condition":
{"type": "NUMERICAL_IS_HIGHER_THAN", "attribute": "fwd_iat_min", "threshold":
13390.0}}]}, {"value": {"type": "PROBABILITY", "distribution":
[0.9377064653138273, 0.06229353468617272], "num_examples": 2119.0}, "condition":
{"type": "NUMERICAL_IS_HIGHER_THAN", "attribute": "fwd_iat_min", "threshold":
10945.0}, "children": [{"value": {"type": "PROBABILITY", "distribution":
[0.9984810126582279, 0.0015189873417721519], "num_examples": 1975.0},
"condition": {"type": "NUMERICAL_IS_HIGHER_THAN", "attribute": "bwd_iat_mean",
"threshold": 98455328.0}}, {"value": {"type": "PROBABILITY", "distribution":
[0.10416666666666667, 0.8958333333333334], "num_examples": 144.0}, "condition":
{"type": "NUMERICAL_IS_HIGHER_THAN", "attribute": "bwd_iat_mean", "threshold":
41052648.0}}]}]}, {"value": {"type": "PROBABILITY", "distribution":
[0.9963651947329722, 0.0036348052670277585], "num_examples": 697699.0},
"condition": {"type": "NUMERICAL_IS_HIGHER_THAN", "attribute": "fwd_iat_mean",
"threshold": 1354163.75}, "children": [{"value": {"type": "PROBABILITY",
"distribution": [0.9840354928450643, 0.015964507154935723], "num_examples":
109994.0}, "condition": {"type": "NUMERICAL_IS_HIGHER_THAN", "attribute":
"flow_duration", "threshold": 3323335.0}, "children": [{"value": {"type":
"PROBABILITY", "distribution": [0.9981127865682464, 0.0018872134317535281],
"num_examples": 107566.0}, "condition": {"type": "NUMERICAL_IS_HIGHER_THAN",
"attribute": "fwd_iat_mean", "threshold": 81613264.0}}, {"value": {"type":
"PROBABILITY", "distribution": [0.3603789126853377, 0.6396210873146623],
"num_examples": 2428.0}, "condition": {"type": "NUMERICAL_IS_HIGHER_THAN",
"attribute": "fwd_iat_min", "threshold": 1069933.0}}]}, {"value": {"type":
"PROBABILITY", "distribution": [0.9986728035323844, 0.0013271964676155553],
"num_examples": 587705.0}, "condition": {"type": "NUMERICAL_IS_HIGHER_THAN",
"attribute": "flow_duration", "threshold": 16.5}, "children": [{"value":
{"type": "PROBABILITY", "distribution": [0.999638447237223,
0.00036155276277707970], "num_examples": 564233.0}, "condition": {"type":
"NUMERICAL_IS_HIGHER_THAN", "attribute": "flow_duration", "threshold":
17297.0}}, {"value": {"type": "PROBABILITY", "distribution":
[0.9754601226993865, 0.024539877300613498], "num_examples": 23472.0},
"condition": {"type": "NUMERICAL_IS_HIGHER_THAN", "attribute": "flow_duration",

```
"threshold": 1.5}}]}]}]}]},
"#tree_plot_bc16f2b0ee554b4ba184df72c01229b0")\n</script>\n'
```

[14]:

```
# Erstellen einer Bilanz für das trainierte Modell
model.summary()
```

```
Model: "random_forest_model"
_________________________________________________________________
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_iat_mean
        flow_duration
        fwd_iat_mean
        fwd_iat_min

No weights

Variable Importance: NUM_NODES:
    1. "flow_duration" 24569.000000 ################
    2.   "fwd_iat_min" 16023.000000 #######
    3.  "fwd_iat_mean" 14166.000000 ######
    4.  "bwd_iat_mean" 7501.000000

Variable Importance: NUM_AS_ROOT:
    1. "bwd_iat_mean" 223.000000 ################
    2. "fwd_iat_mean" 77.000000

Variable Importance: SUM_SCORE:
    1.  "bwd_iat_mean" 6081381.699483 ################
    2. "flow_duration" 2403939.750272 #
    3.  "fwd_iat_mean" 2038630.496041
    4.   "fwd_iat_min" 1933959.961162

Variable Importance: MEAN_MIN_DEPTH:
    1.       "__LABEL" 10.152501 ################
    2.   "fwd_iat_min"  3.473030 ####
    3.  "fwd_iat_mean"  3.062956 ###
    4. "flow_duration"  2.615873 ###
    5.  "bwd_iat_mean"  0.850565



Winner take all: true
Out-of-bag evaluation: accuracy:0.999464 logloss:0.0117982
Number of trees: 300
Total number of nodes: 124818

Number of nodes by tree:
Count: 300 Average: 416.06 StdDev: 21.7356
Min: 349 Max: 471 Ignored: 0
----------------------------------------------
[ 349, 355)  1   0.33%   0.33%
[ 355, 361)  1   0.33%   0.67%
[ 361, 367)  2   0.67%   1.33%
[ 367, 373)  2   0.67%   2.00%
[ 373, 379)  7   2.33%   4.33% #
[ 379, 385) 12   4.00%   8.33% ##
[ 385, 392) 11   3.67%  12.00% ##
[ 392, 398) 24   8.00%  20.00% #####
[ 398, 404) 20   6.67%  26.67% ####
[ 404, 410) 28   9.33%  36.00% #####
[ 410, 416) 53  17.67%  53.67% ##########
[ 416, 422) 31  10.33%  64.00% ######
[ 422, 428) 24   8.00%  72.00% #####
[ 428, 435) 24   8.00%  80.00% #####
[ 435, 441) 17   5.67%  85.67% ###
[ 441, 447) 11   3.67%  89.33% ##
[ 447, 453) 17   5.67%  95.00% ###
[ 453, 459)  6   2.00%  97.00% #
[ 459, 465)  5   1.67%  98.67% #
[ 465, 471]  4   1.33% 100.00% #

Depth by leafs:
Count: 62559 Average: 10.1512 StdDev: 2.7814
Min: 2 Max: 15 Ignored: 0
----------------------------------------------
[  2,  3)    7   0.01%   0.01%
[  3,  4)  183   0.29%   0.30%
[  4,  5) 1140   1.82%   2.13% #
[  5,  6) 1881   3.01%   5.13% ##
[  6,  7) 3443   5.50%  10.64% ####
[  7,  8) 4624   7.39%  18.03% ######
[  8,  9) 6281  10.04%  28.07% ########
[  9, 10) 8346  13.34%  41.41% ##########
[ 10, 11) 8301  13.27%  54.68% ##########
[ 11, 12) 7756  12.40%  67.08% #########
[ 12, 13) 6771  10.82%  77.90% ########
[ 13, 14) 5393   8.62%  86.52% ######
[ 14, 15) 4051   6.48%  93.00% #####
[ 15, 15] 4382   7.00% 100.00% #####

Number of training obs by leaf:
Count: 62559 Average: 3383.29 StdDev: 20312.3
Min: 5 Max: 313321 Ignored: 0
----------------------------------------------
[      5,  15670) 60484  96.68%  96.68% ##########
[  15670,  31336)   704   1.13%  97.81%
[  31336,  47002)   254   0.41%  98.21%
[  47002,  62668)   174   0.28%  98.49%
[  62668,  78334)    84   0.13%  98.63%
[  78334,  94000)   210   0.34%  98.96%
[  94000, 109665)    45   0.07%  99.03%
[ 109665, 125331)    50   0.08%  99.11%
[ 125331, 140997)   123   0.20%  99.31%
[ 140997, 156663)    93   0.15%  99.46%
[ 156663, 172329)    37   0.06%  99.52%
[ 172329, 187995)    21   0.03%  99.55%
[ 187995, 203661)    26   0.04%  99.59%
[ 203661, 219326)    74   0.12%  99.71%
[ 219326, 234992)    71   0.11%  99.83%
[ 234992, 250658)    41   0.07%  99.89%
[ 250658, 266324)     7   0.01%  99.90%
[ 266324, 281990)     8   0.01%  99.92%
[ 281990, 297656)    39   0.06%  99.98%
[ 297656, 313321]    14   0.02% 100.00%

Attribute in nodes:
        24569 : flow_duration [NUMERICAL]
        16023 : fwd_iat_min [NUMERICAL]
        14166 : fwd_iat_mean [NUMERICAL]
        7501 : bwd_iat_mean [NUMERICAL]

Attribute in nodes with depth <= 0:
        223 : bwd_iat_mean [NUMERICAL]
        77 : fwd_iat_mean [NUMERICAL]

Attribute in nodes with depth <= 1:
        283 : bwd_iat_mean [NUMERICAL]
        229 : flow_duration [NUMERICAL]
        224 : fwd_iat_min [NUMERICAL]
        164 : fwd_iat_mean [NUMERICAL]

Attribute in nodes with depth <= 2:
        660 : flow_duration [NUMERICAL]
        544 : bwd_iat_mean [NUMERICAL]
        499 : fwd_iat_min [NUMERICAL]
        390 : fwd_iat_mean [NUMERICAL]

Attribute in nodes with depth <= 3:
        1198 : flow_duration [NUMERICAL]
        1118 : fwd_iat_min [NUMERICAL]
        1037 : bwd_iat_mean [NUMERICAL]
        943 : fwd_iat_mean [NUMERICAL]

Attribute in nodes with depth <= 5:
        4001 : flow_duration [NUMERICAL]
        3014 : fwd_iat_min [NUMERICAL]
        2798 : fwd_iat_mean [NUMERICAL]
        2400 : bwd_iat_mean [NUMERICAL]

Condition type in nodes:
        62259 : 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:
        2093 : HigherCondition
Condition type in nodes with depth <= 3:
        4296 : HigherCondition
Condition type in nodes with depth <= 5:
        12213 : HigherCondition
Node format: NOT_SET

Training OOB:
        trees: 1, Out-of-bag evaluation: accuracy:0.999245 logloss:0.0272256
        trees: 11, Out-of-bag evaluation: accuracy:0.999411 logloss:0.0160011
        trees: 21, Out-of-bag evaluation: accuracy:0.999436 logloss:0.0143874
        trees: 31, Out-of-bag evaluation: accuracy:0.999456 logloss:0.0136232
        trees: 41, Out-of-bag evaluation: accuracy:0.99945 logloss:0.0128219
        trees: 51, Out-of-bag evaluation: accuracy:0.999447 logloss:0.0127773
        trees: 61, Out-of-bag evaluation: accuracy:0.999443 logloss:0.012591
        trees: 71, Out-of-bag evaluation: accuracy:0.999453 logloss:0.012499
        trees: 81, Out-of-bag evaluation: accuracy:0.999456 logloss:0.0124896
        trees: 91, Out-of-bag evaluation: accuracy:0.999454 logloss:0.0124424
        trees: 101, Out-of-bag evaluation: accuracy:0.999459 logloss:0.0123936
        trees: 111, Out-of-bag evaluation: accuracy:0.999457 logloss:0.0122594
        trees: 121, Out-of-bag evaluation: accuracy:0.999456 logloss:0.0121669
        trees: 131, Out-of-bag evaluation: accuracy:0.999459 logloss:0.0121653
        trees: 141, Out-of-bag evaluation: accuracy:0.999459 logloss:0.0120756
        trees: 151, Out-of-bag evaluation: accuracy:0.999459 logloss:0.0120764
        trees: 161, Out-of-bag evaluation: accuracy:0.999463 logloss:0.0120317
```

```
trees: 171, Out-of-bag evaluation: accuracy:0.999464 logloss:0.0118936
trees: 181, Out-of-bag evaluation: accuracy:0.99946 logloss:0.0118927
trees: 191, Out-of-bag evaluation: accuracy:0.999461 logloss:0.0118931
trees: 201, Out-of-bag evaluation: accuracy:0.999463 logloss:0.0118914
trees: 211, Out-of-bag evaluation: accuracy:0.999463 logloss:0.0118918
trees: 221, Out-of-bag evaluation: accuracy:0.999461 logloss:0.0118918
trees: 231, Out-of-bag evaluation: accuracy:0.999463 logloss:0.0118431
trees: 241, Out-of-bag evaluation: accuracy:0.999461 logloss:0.0118401
trees: 251, Out-of-bag evaluation: accuracy:0.999464 logloss:0.0118391
trees: 261, Out-of-bag evaluation: accuracy:0.999463 logloss:0.0118403
trees: 271, Out-of-bag evaluation: accuracy:0.999466 logloss:0.011841
trees: 281, Out-of-bag evaluation: accuracy:0.999467 logloss:0.0117971
trees: 291, Out-of-bag evaluation: accuracy:0.999466 logloss:0.0117986
trees: 300, Out-of-bag evaluation: accuracy:0.999464 logloss:0.0117982
```

[15]:

```python
# 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/Slowloris_Model.png')
plt.clf()
```

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