B.I PROJECT

APPLICATION OF NN TO CALIBRATED RANGEFINDER APPROXIMATION

AGNIESZKA GÓRAL (AGNIGOR672@STUDENT.POLSL.PL)
DAMIAN WIECZOREK (DAMIWIE617@STUDENT.POLSL.PL)

PRESENTATION PLAN

1 - OVERVIEW OF THE PROBLEM

2 - DATASET OVERVIEW

3 - ABOUT NEURAL NETWORK

4 - MODEL OVERVIEW

5 - TRAINED MODELS PLOT AND CONCLUSIONS

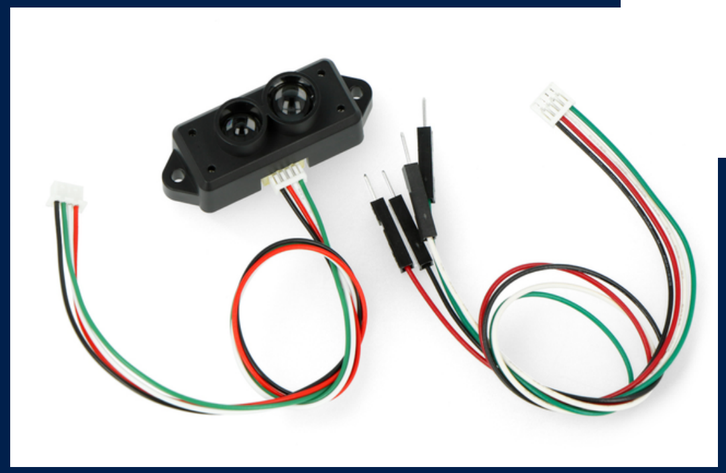
PROBLEM

The goal is to implement the method of calibrating measurements from MiniLidarS in such a way that they are as close as possible to the results of measurements from Leica.

In addition, the measurement error for the proposed method should be determined.

DEVICES

- We received measurements from two devices - Leica and MiniLidarS
- Mini Lidar S - laser meter
- Leica - probably laser meter



MINI LIDAR S

LEICA

DATA SET

- We received the data in a .csv file
- Screen only shows top of the file
- In file is 1920 measurements
- For each Leica value is 40 MiniLidarS values
- We couldn't normalize data, because we didn't know devices used to measure

	A
1	Leica,MiniLidarS
2	0.055,0.082
3	0.055,0.084
4	0.055,0.081
5	0.055,0.082
6	0.055,0.086
7	0.055,0.082
8	0.055,0.086
9	0.055,0.085
10	0.055,0.085
11	0.055,0.081
12	0.055,0.085
13	0.055,0.083
14	0.055,0.083
15	0.055,0.085

DATA SETS PREPARED:

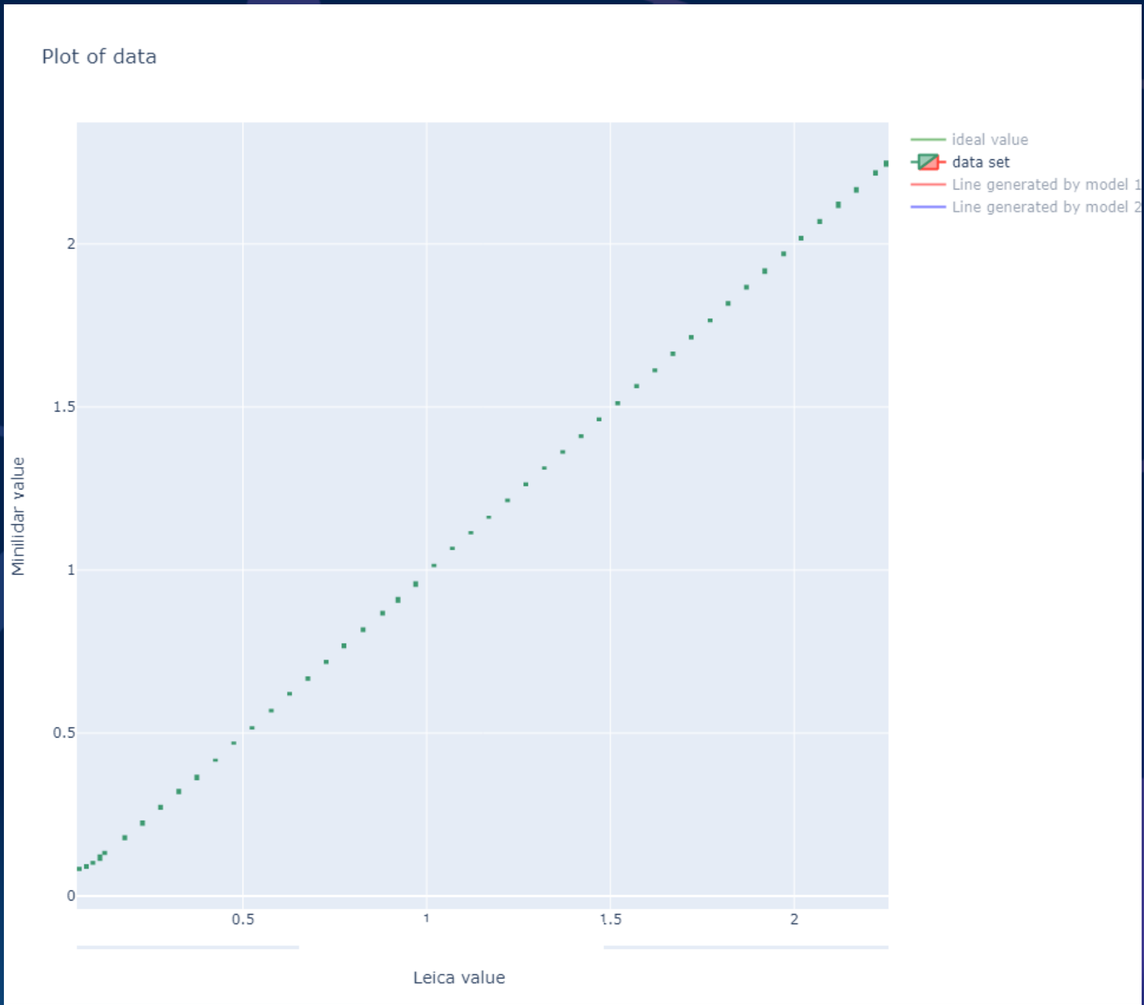
- default dataset without changes (number 1)
- 5 first values for each measurement (number 2)
- 5 closest values to average (number 3)
- dataset sorted ascending (number 4)
- dataset sorted descending (number 5)

DATA SET ANALYSIS

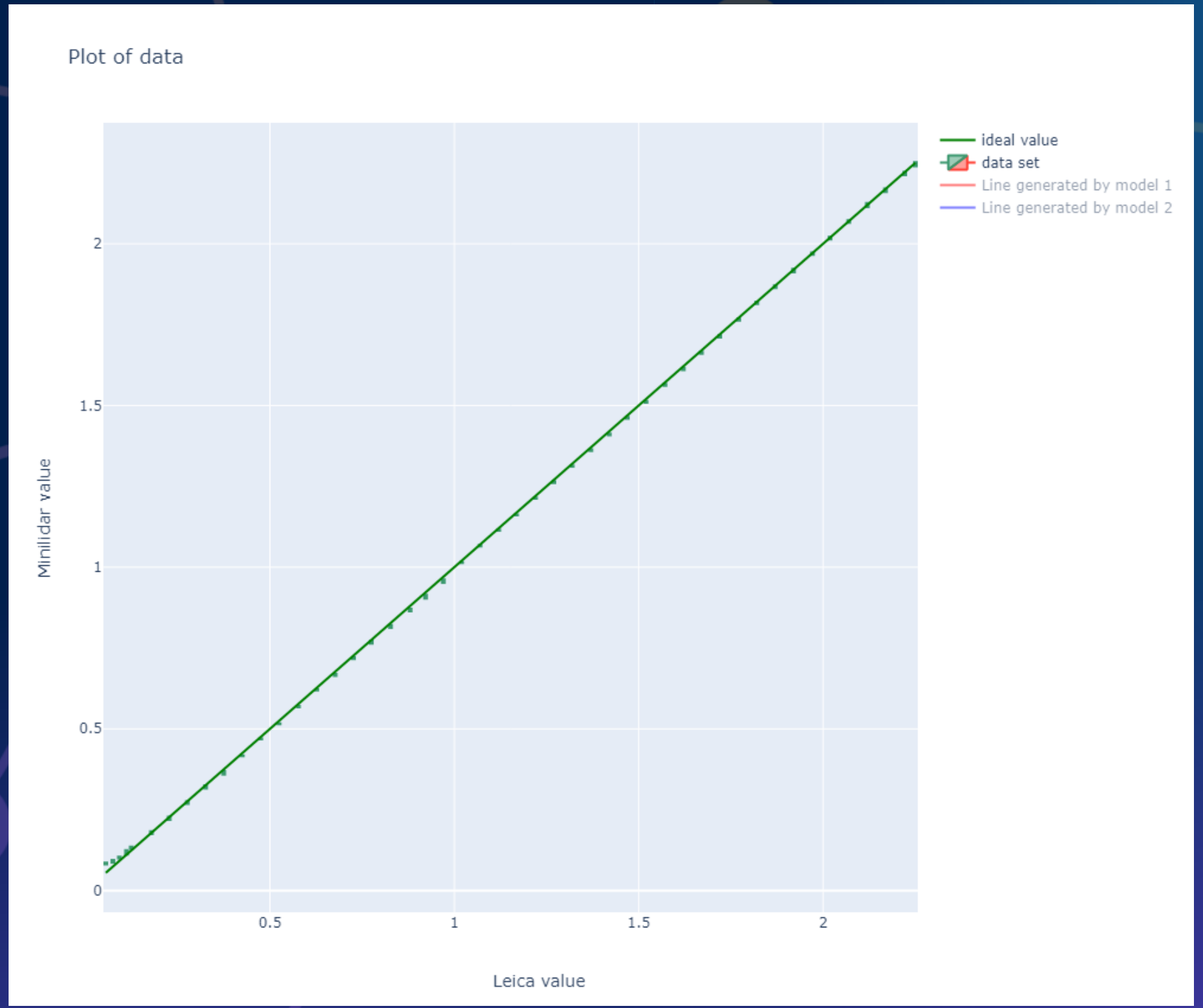
- Analyzing the data set, it turned out that the MiniLidarS measurement values are overestimated for small values, and similar for larger values

CANDLE PLOT

Raw values from dataset



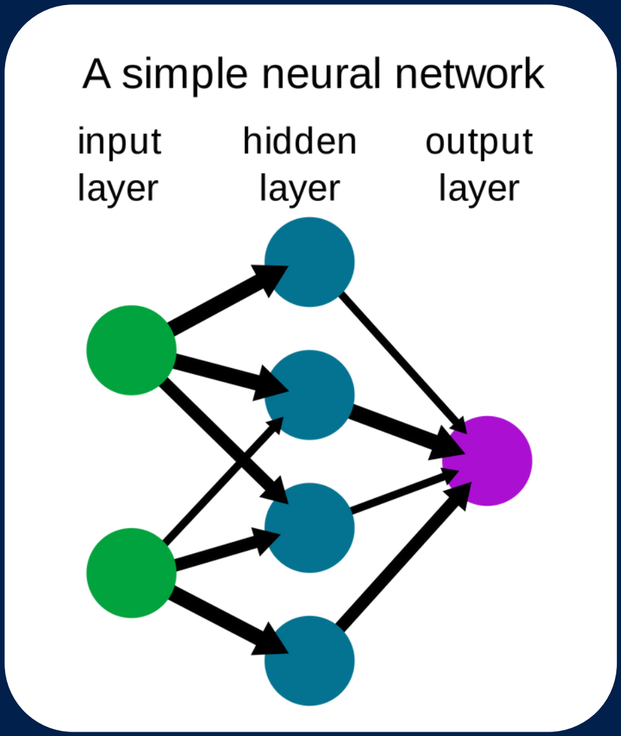
Ideal (Target) value



NEURAL NETWORK

- A neural network is a combination of elements called artificial neurons that create at least three layers: input, hidden and output, with many hidden layers.

NEURAL NETWORK



- **The neural network is divided into three layers: input, hidden and output.**
- **The input and output layers depend on the format of the input and expected data.**

NEURAL NETWORK

- **The hidden layer depends on what we want to achieve with a given network.**

- **There can be many hidden layers. The more of them, the more complex problems we can solve. A single hidden layer can solve a linear problem, but two hidden layers can solve a quadratic problem and so on.**

OUR NEURAL NETWORK

- We used the Multi-layer Perceptron regressor from the sklearn library
- The same model adjusts the number of neurons in the input and output layers.
- We created two models with different number of hidden layers to see the difference in the results

OUR NEURAL NETWORK

- **The first model has 10 hidden layers with the number of neurons from 100 to 10 decreasing by 10**

- **The second model has only 3 hidden layers with 100, 50 and 10 neurons**

PYTHON LIBRARIES:

SKLEARN

PLOTLY

PANDAS

NUMPY

PYTHON LIBRARIES:

SKLEARN:

- model of NN
- functions to calculate accuracy of NN

PLOTLY:

- create and show interactive plots

PANDAS:

- used to feed data to plotly

NUMPY:

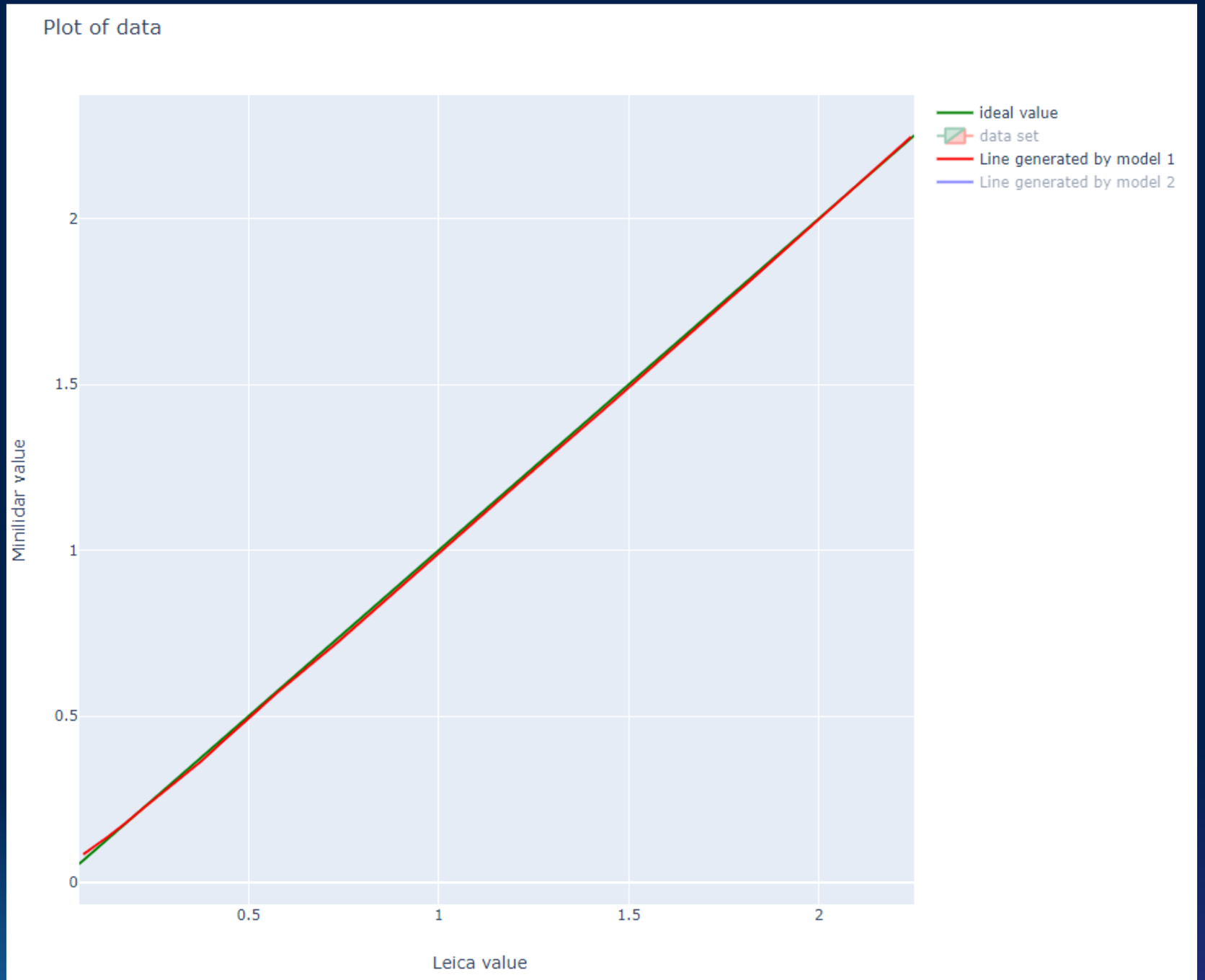
- used to feed data to sklearn model

ABOUT IMPLEMENTATION

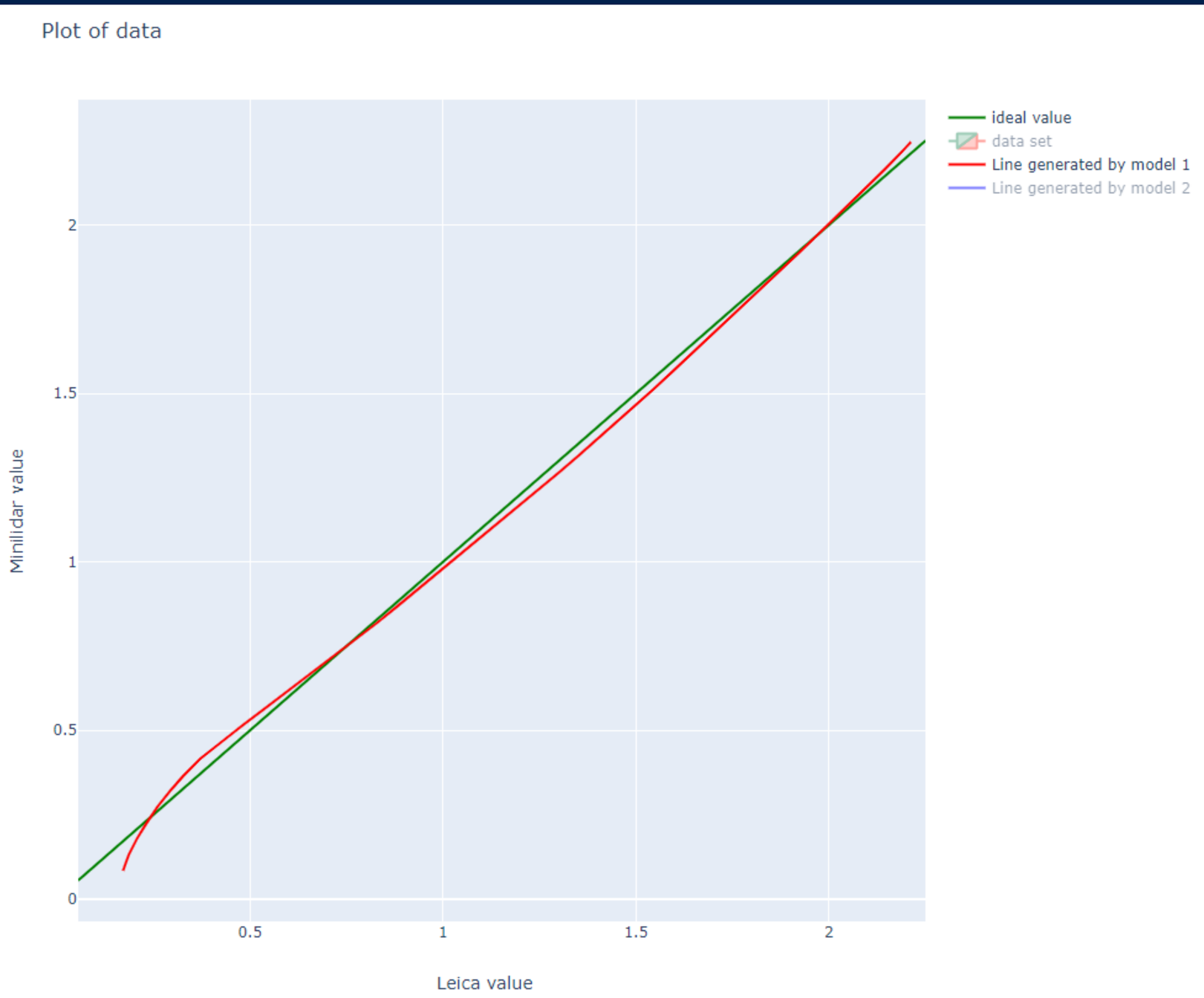
- All datasets were divided into test and training sets with a 20/80 division
- The model was exercised on a training dataset, but the mean squared error of the model predictions was checked with a test set
- The accuracy of the model was checked using both the mean squared error and the score method from the sklearn library

RESUSLTS WE RECEIVED

MSE - mean square error in meters



Result for 1st model
Dataset: 1
Accuracy: 99.9382%
MSE: 0.000014 m

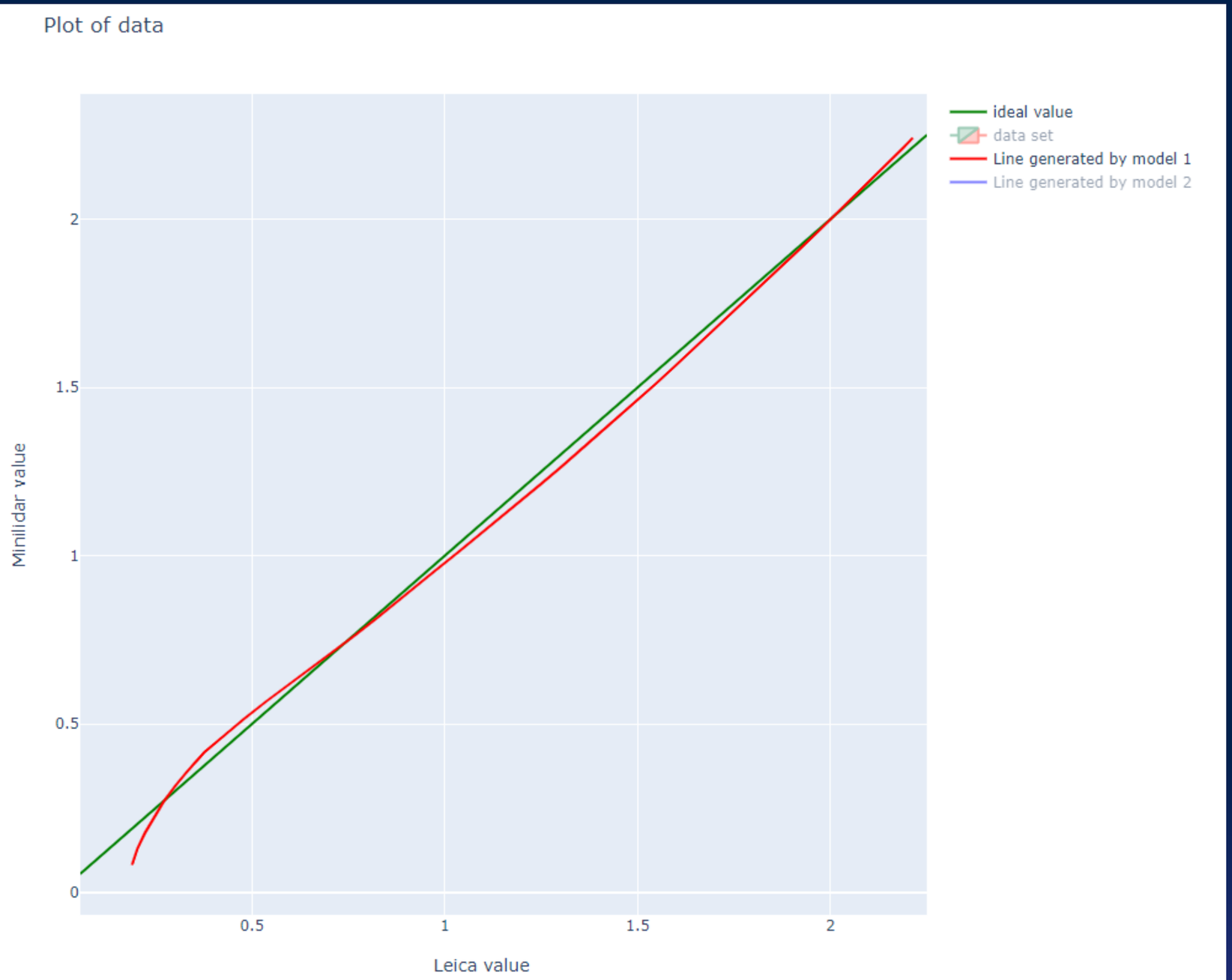


Result for 1st model

Dataset: 2

Accuracy: 98.6320%

MSE: 0.00154 m

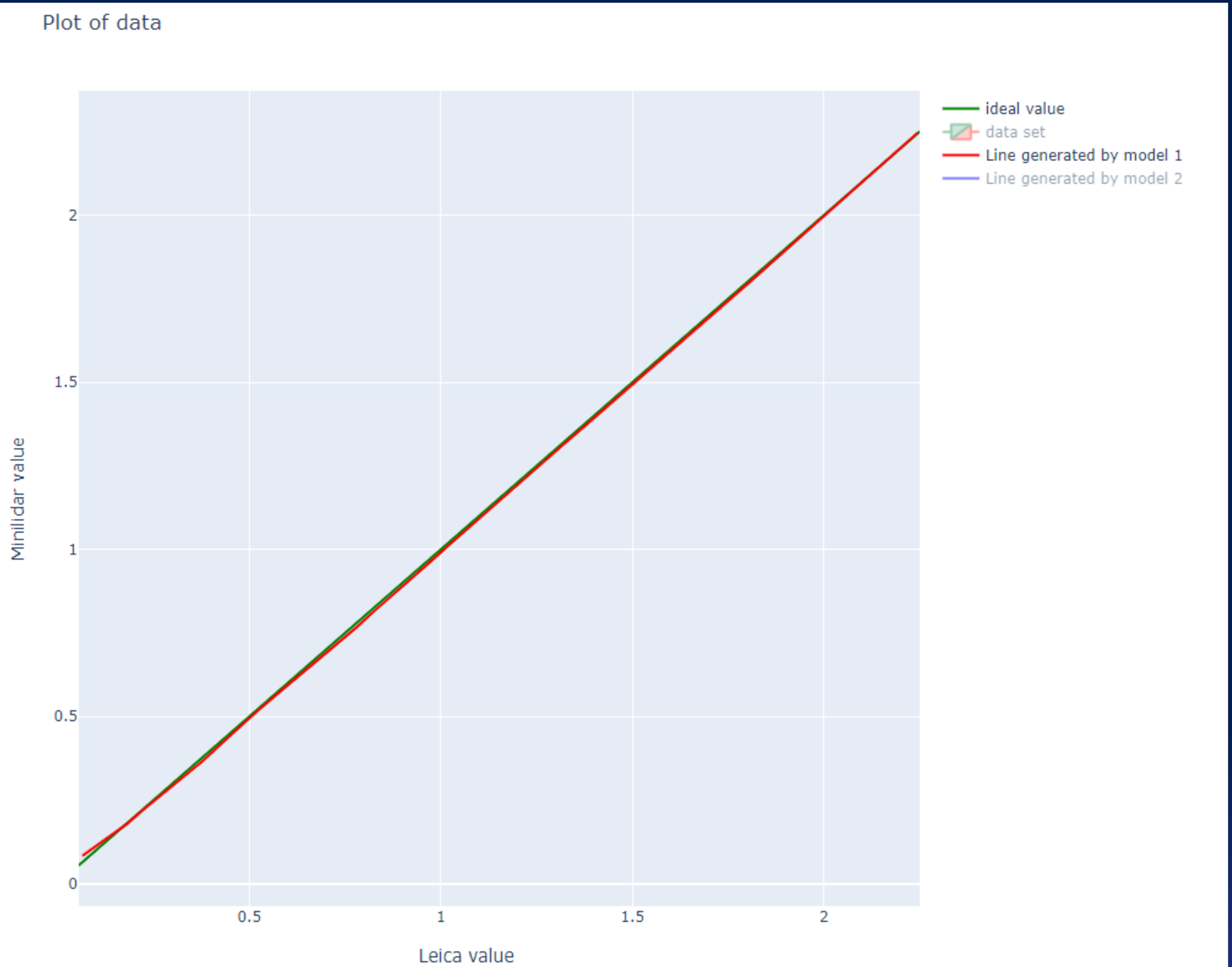


Result for 1st model

Dataset: 3

Accuracy: 98.5394%

MSE: 0.002074 m

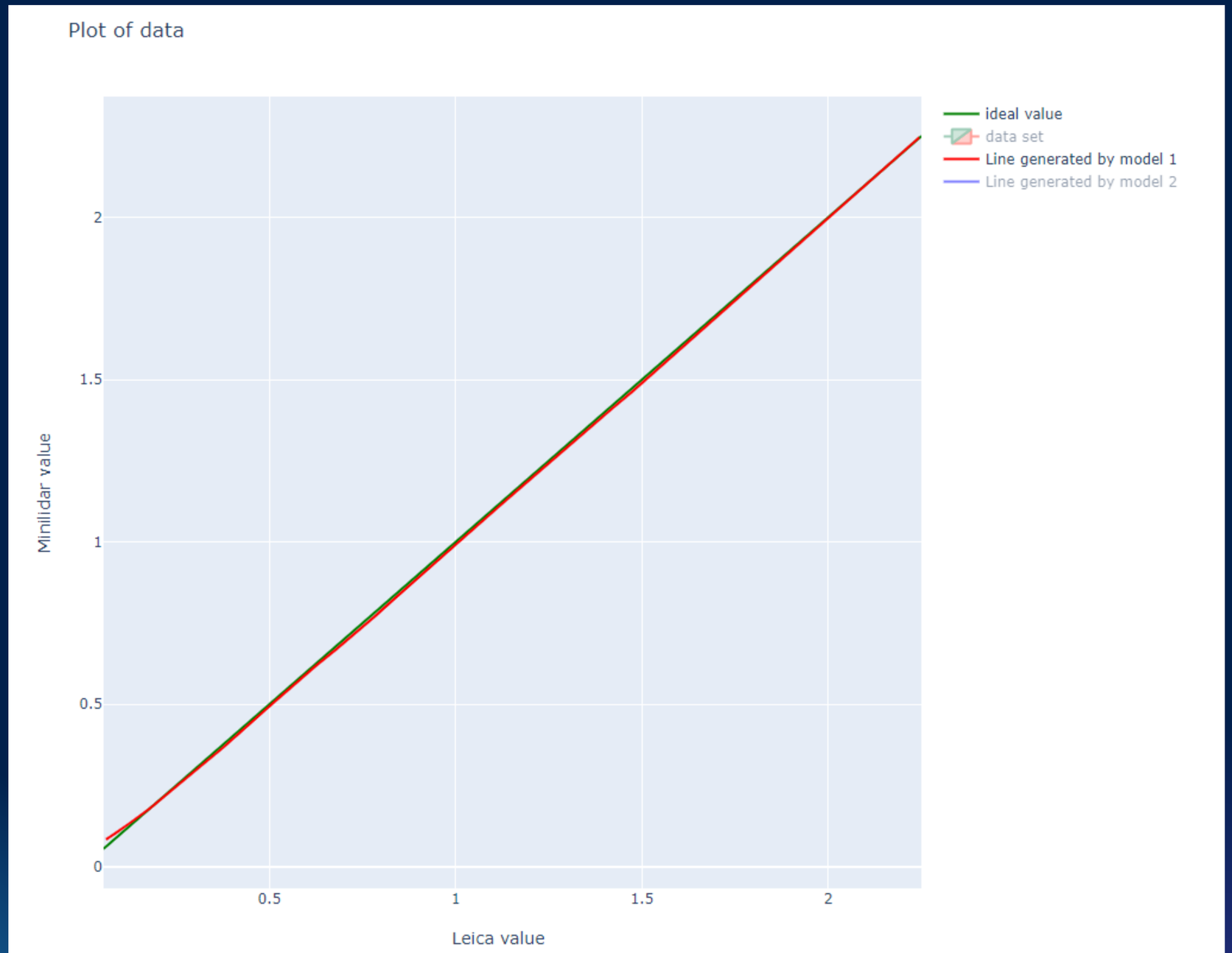


Result for 1st model

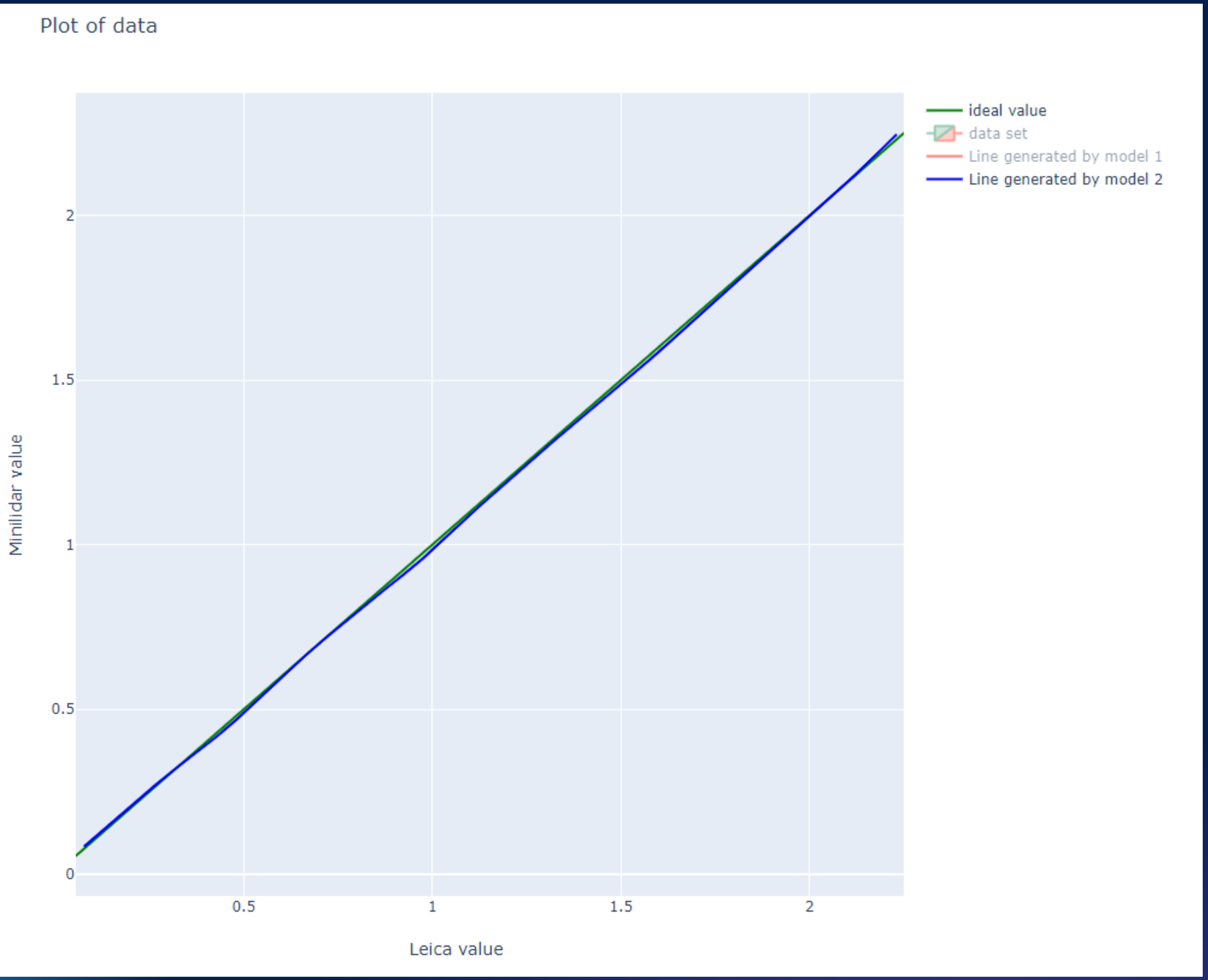
Dataset: 4

Accuracy: 99.94%

MSE: 0.000011 m



Result for 1st model
Dataset: 5
Accuracy: 99.9368%
MSE: 0.000011 m

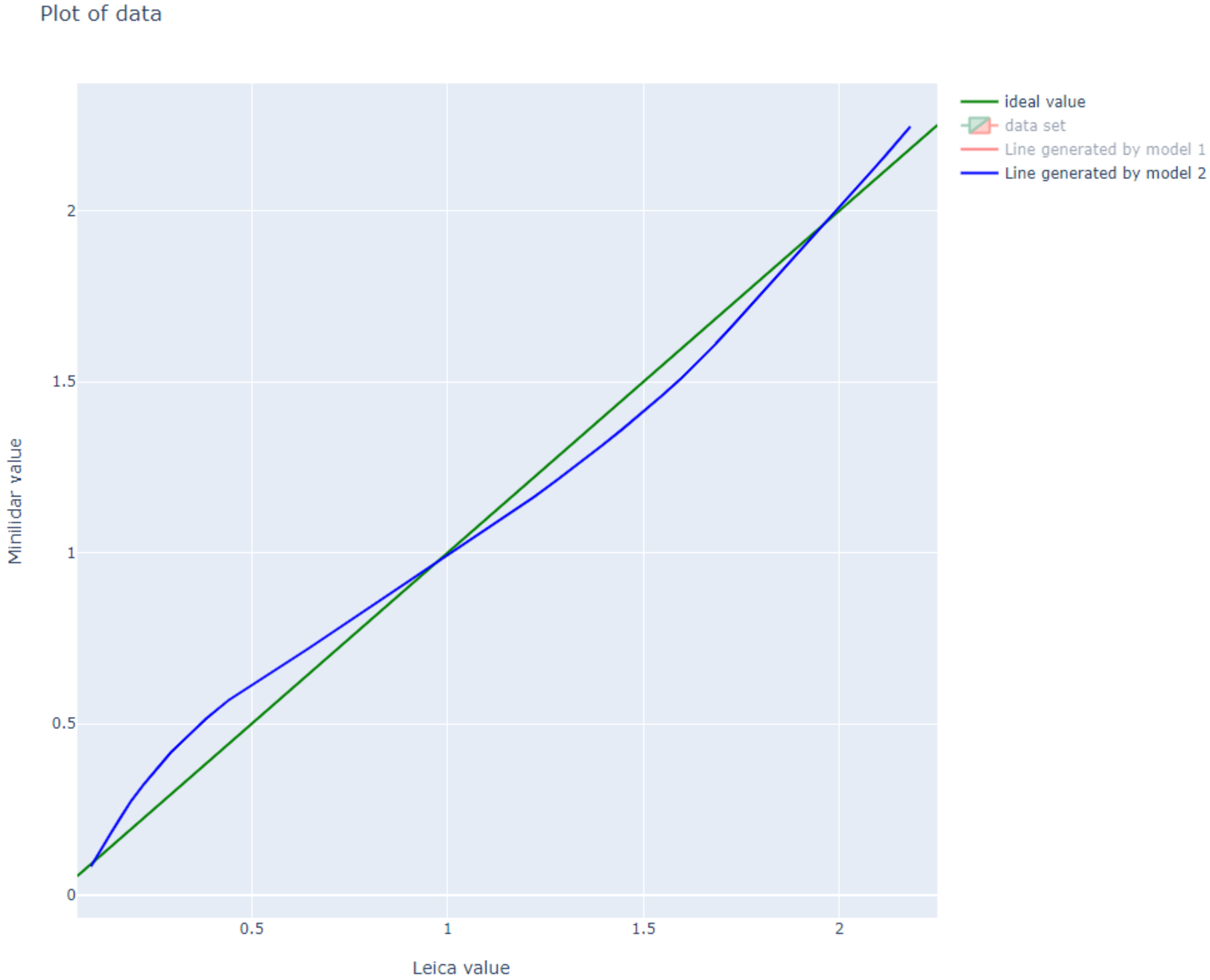


Result for 2nd model

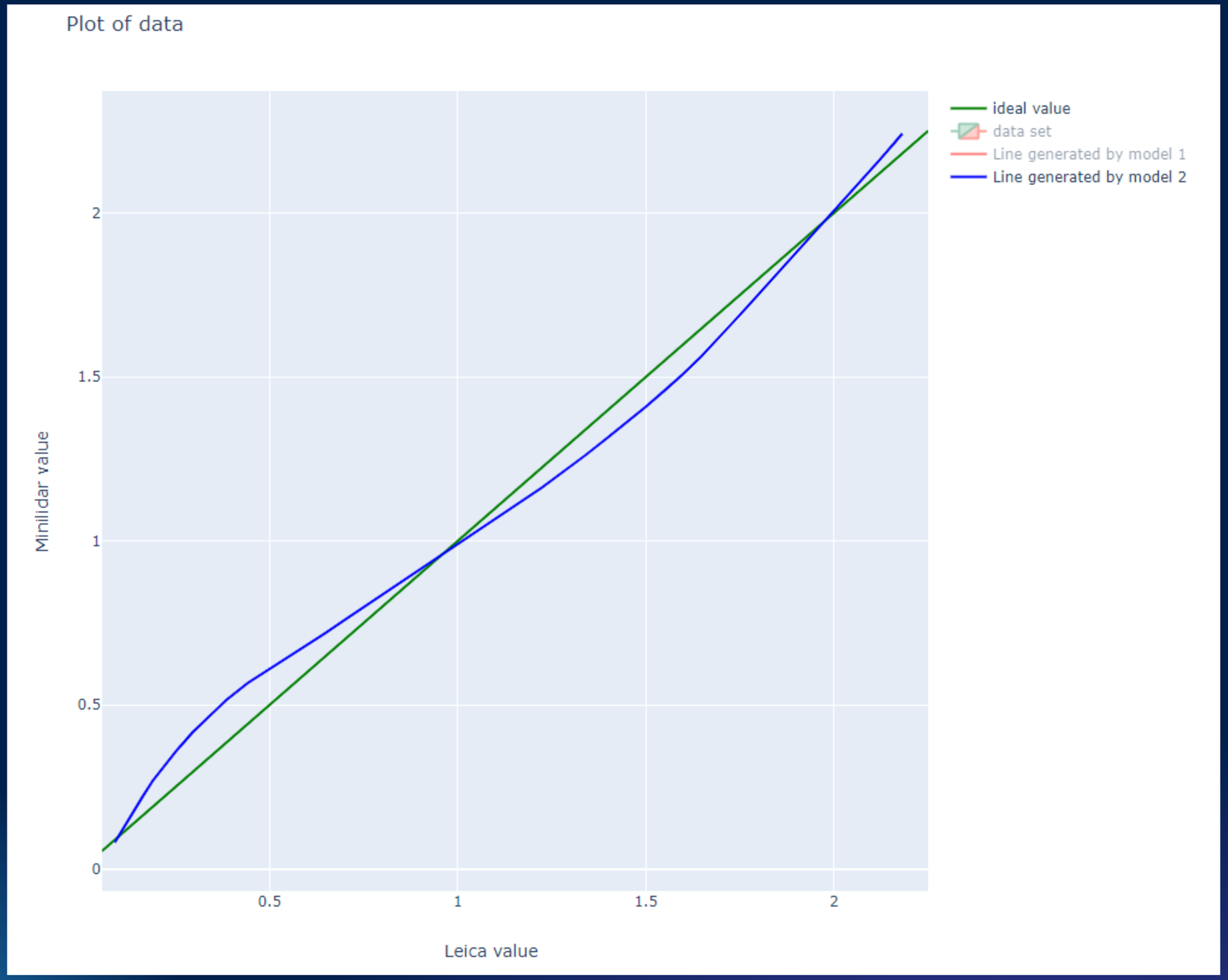
Dataset: 1

Accuracy: 99.9278%

MSE: 0.00005 m



Result for 2nd model
Dataset: 2
Accuracy: 97.2831%
MSE: 0.003971 m

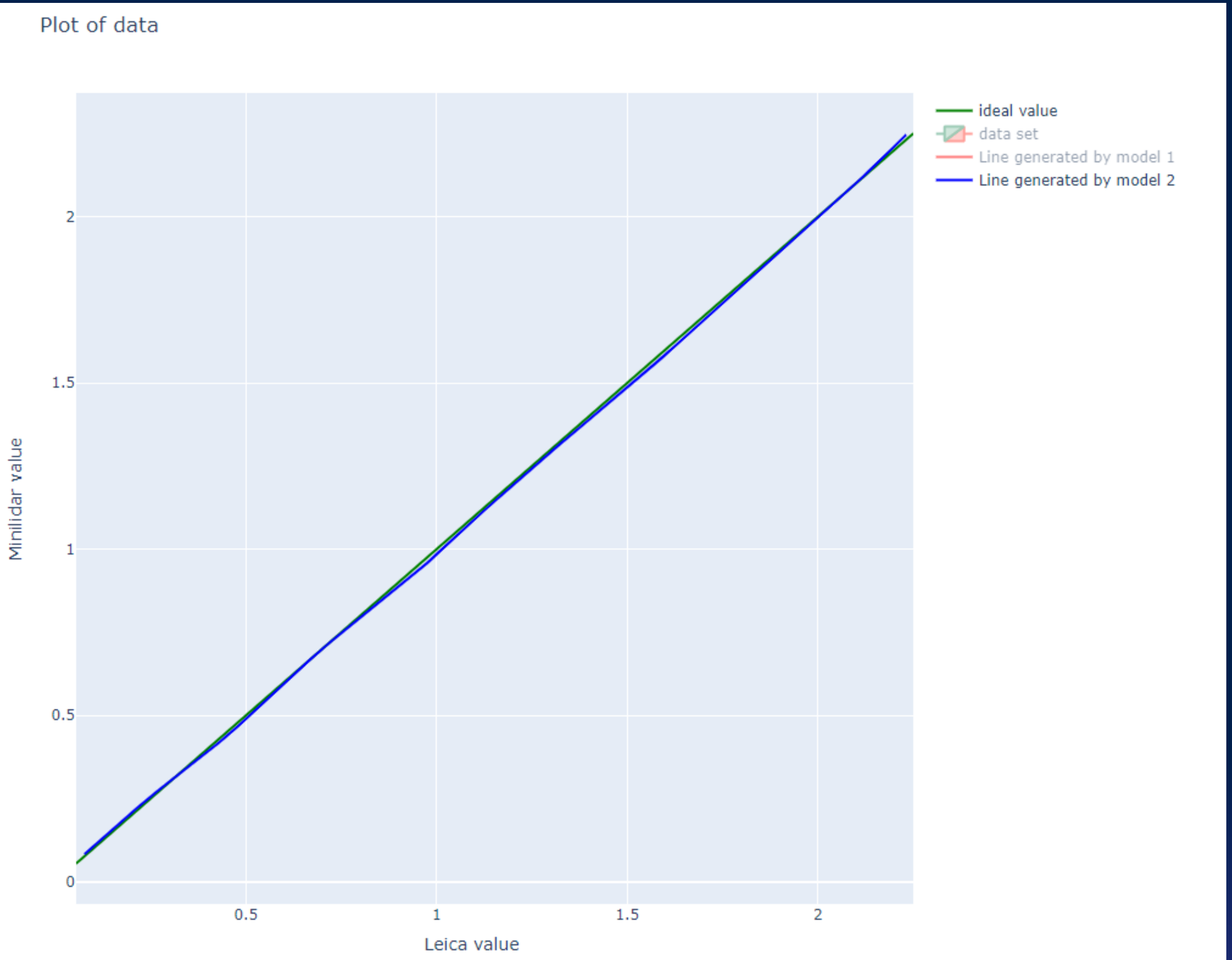


Result for 2nd model

Dataset: 3

Accuracy: 97.2937%

MSE 0.003979 m

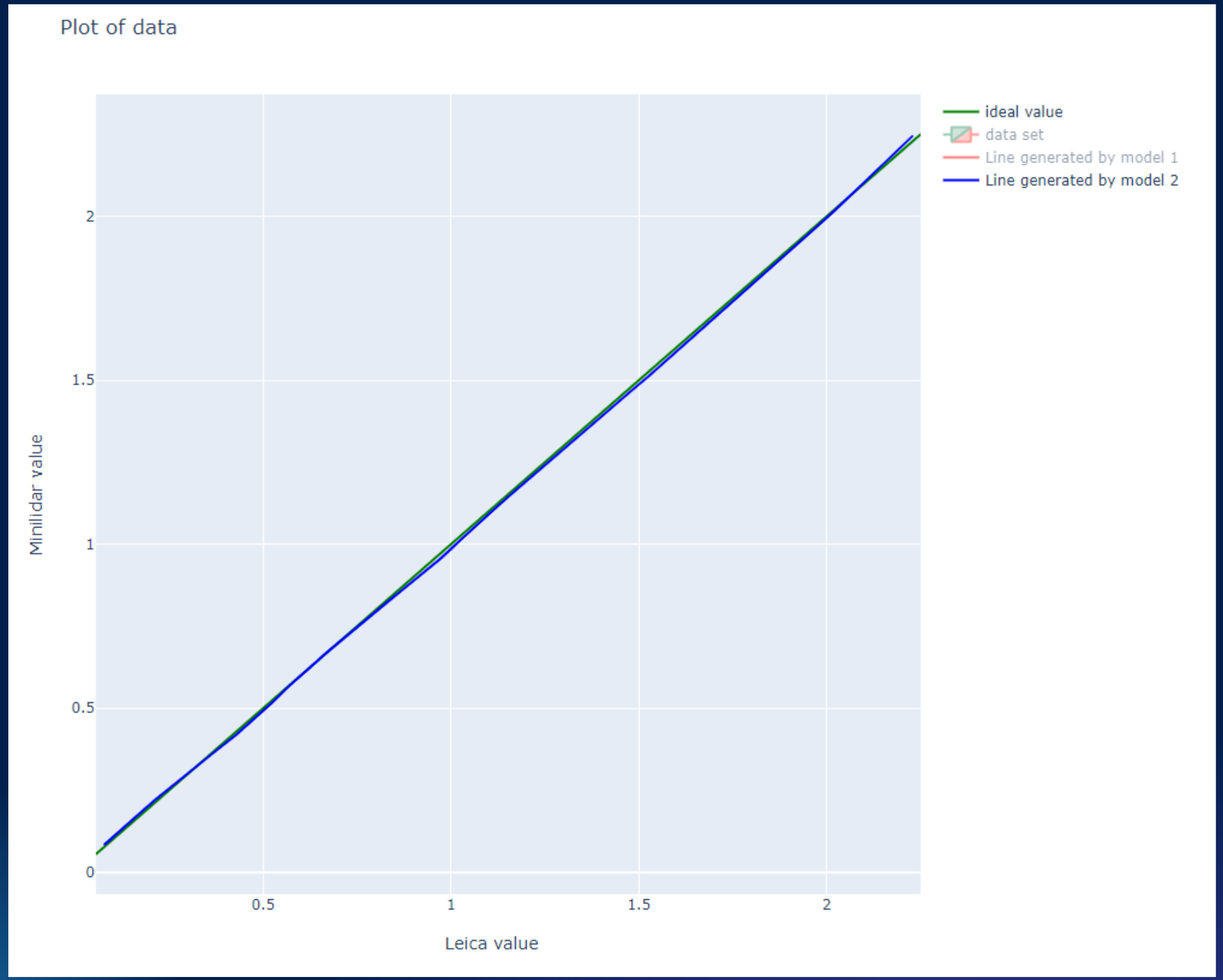


Result for 2nd model

Dataset: 4

Accuracy: 99.9256%

MSE: 0.000051 m



Result for 2nd model

Dataset: 5

Accuracy: 99.9316%

MSE: 0.000054 m

DEFAULT DATASET WITHOUT CHANGES

DATASET 1	MODEL 1	MODEL 2
ACCURACY	99,9382%	99,9278%
MSE	0.000014	0,00005

5 FIRST VALUES FOR EACH MEASUREMENT

DATASET 2	MODEL 1	MODEL 2
ACCURACY	98,632%	97,2831%
MSE	0,00154	0,003971

5 CLOSEST VALUES TO AVERAGE

DATASET 3	MODEL 1	MODEL 2
ACCURACY	98,5394%	97,2937%
MSE	0,002074	0,003979

MSE - mean square error

DATASET SORTED ASCENDING

DATASET 4	MODEL 1	MODEL 2
ACCURACY	99,94%	99,9256%
MSE	0.000011	0,000051

DATASET SORTED DESCENDING

DATASET 5	MODEL 1	MODEL 2
ACCURACY	99,9368%	99,9316%
MSE	0.000011	0,000054

- **The second model fared much worse, due to the smaller number of hidden layers.**
- **Model 1 obtained the highest accuracy for all received data sorted in ascending order.**
- **Model 1 performed better for data sorted in ascending order than for data sorted in descending order.**

MSE - mean square error in meters

AN EXAMPLE OF HOW THE PROGRAM WORKS

LEICA REFERENCE VALUE [m]	0,055	0,227	0,525	0,827	1,12	1,42	1,72	2,019
MINI LIDAR S VALUE [m]	0,081	0,219	0,515	0,813	1,113	1,408	1,711	2,014
RESULT VALUE [m]	0,062	0,221	0,52	0,824	1,121	1,415	1,719	1,016

FOR CHOOSEN MODEL WITH BEST PERFORMANCE - MODEL 1, TRAINING SET: 4

THANK YOU FOR YOUR ATTENTION

Q&A TIME

AGNIESZKA GÓRAL (AGNIGOR672@STUDENT.POLSL.PL)
DAMIAN WIECZOREK (DAMIWIE617@STUDENT.POLSL.PL)