NETWORK MEASUREMENT AND DATA ANALYSIS LAB

# PROJECT #4:

## FAILURE DETECTION WITH DIFFERENT FAILURES LOCATION

Bernazzoli Viola - Fiorentino Giuseppe - Roumpeas Konstantinos

# Problem Overview

The scenario considered, was composed by three lightpaths and a monitoring device located at the end node.

The monitoring device captured eight scenarios characterized by two failures:
- excessive attenuation
- extra filtering

Based on the datasets provided, it is needed to be identified if a failure is a present in a lightpath or not

# Topics covered 1/2

- Original vs normalized data accuracy
- Characterization of each classifier based on: accuracy score, runtime and trained model size
- Examine what is the impact of window duration and windows spacing on the performance of each classifier

# Topics covered
## 2/2

- Manually Identify the importance of each feature on the performance and exploit tecniques that can improve the accuracy of the classifier
- Examine the accuracy score of each classifier, when only one failure co-exists in the dataset with the flawless Scenario 0
- XAI: use explainability to realize each feature's role in the Random Forest classifier model

# Workaround

It has been decided to use three different classifiers:
- K Neighbours
- Decision Tree
- Neural Network

Each one has different accuracy, size of the model and runtime.

**KNN** 6 Closest Neighbours

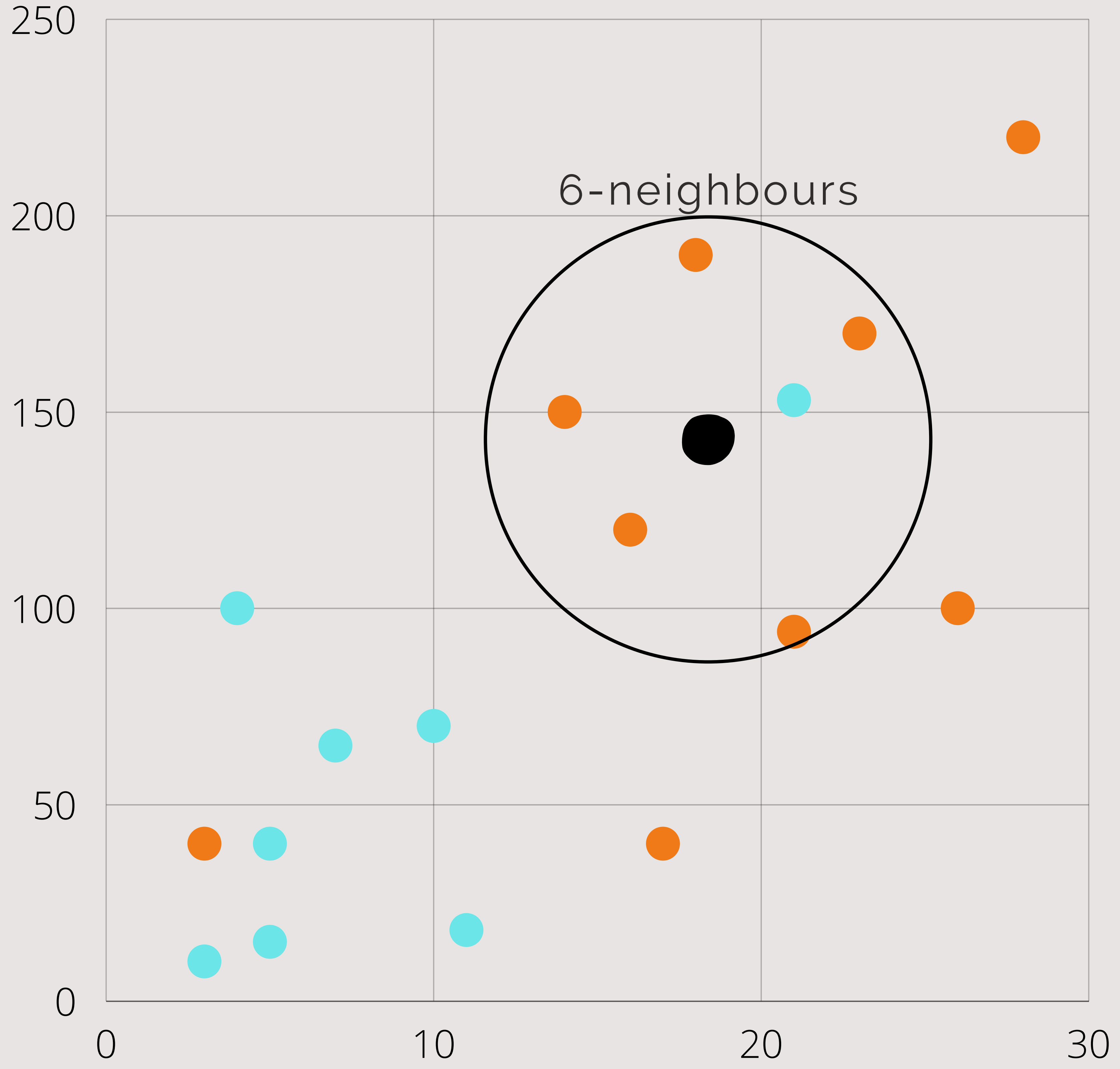
**XGB** eXtreme Gradient Boosting

**MLP** MultiLayer Perceptron

# KNN

TRAINING:
Each point is represented in an N-dimensional space (every dimension is a different feature).

TESTING:
K closest neighbours are considered from the point to be labelled. The groundtruth of the objective point is decided based on majority



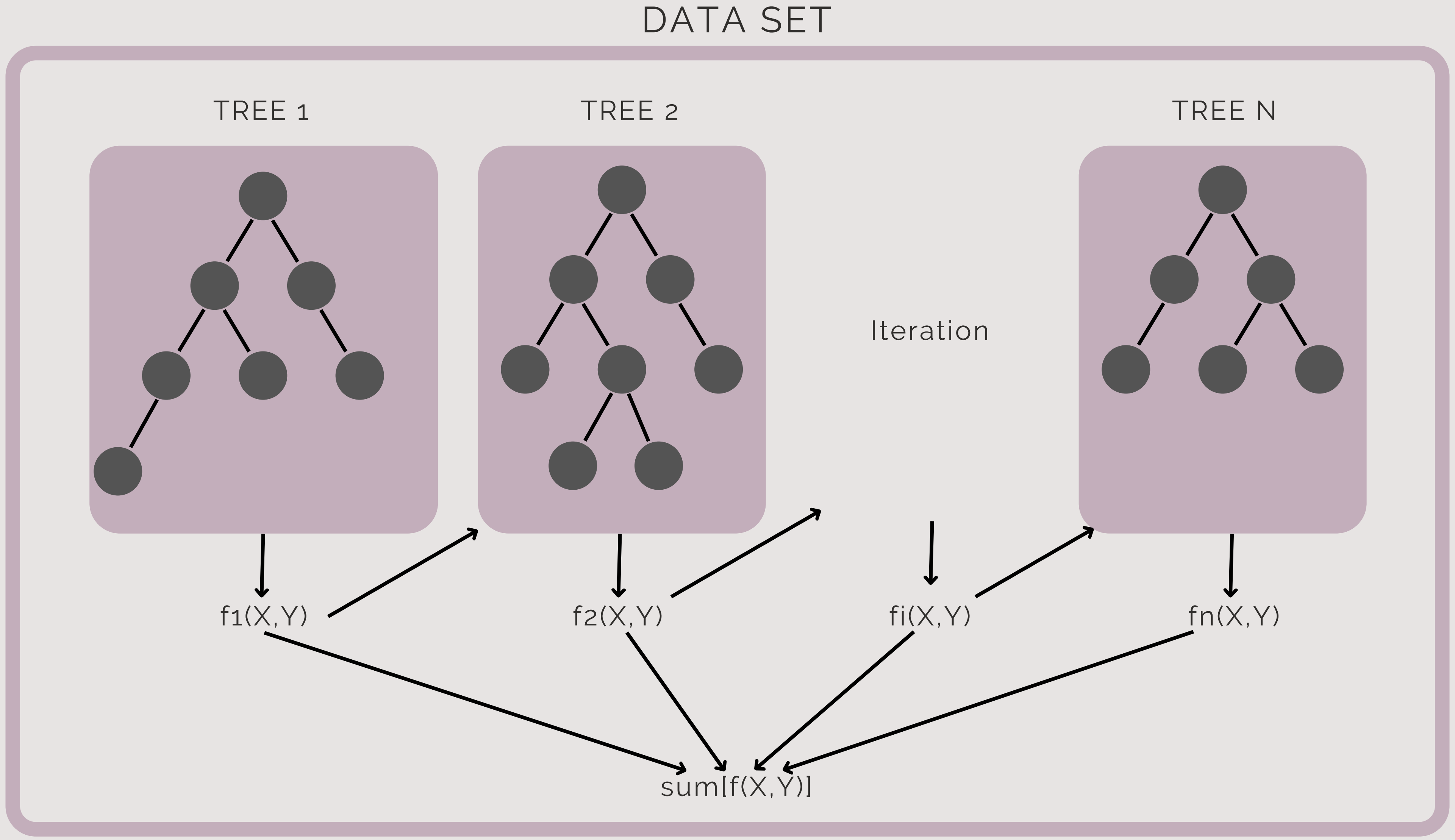

In the picture: ⬤ dot to be analysed -> 6-neighbours (1 blue, 5 orange), the point will be labelled orange

# XGB

Decision tree classifier based on descendant gradient algorithm

TRAINING:
Iterative evaluation process which decreses the residuals at each iteration. Resulting in a sequence of trees.

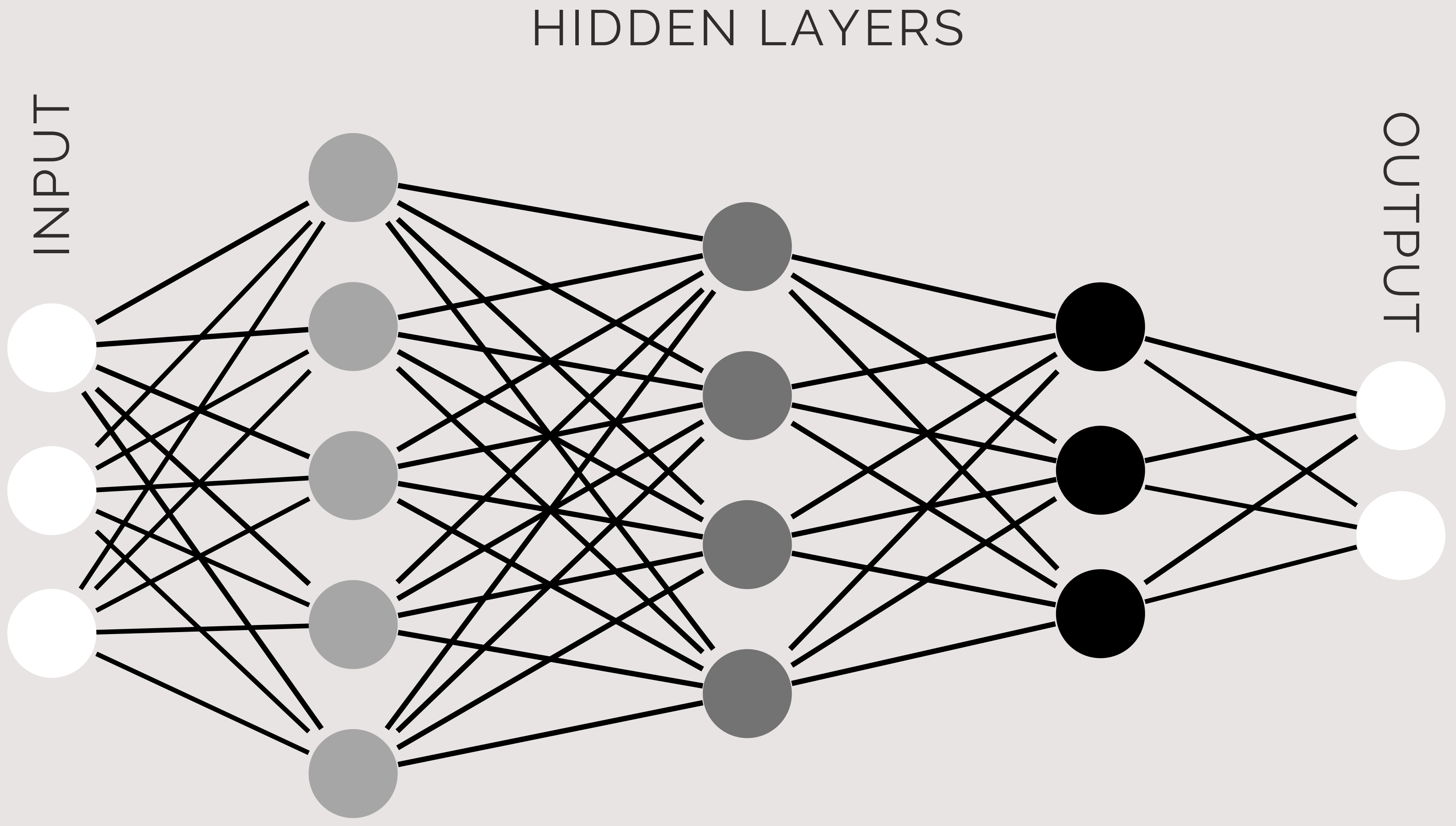
TESTING:
The prediction scores of each individual tree are weighted and summed up to get the prediction value.



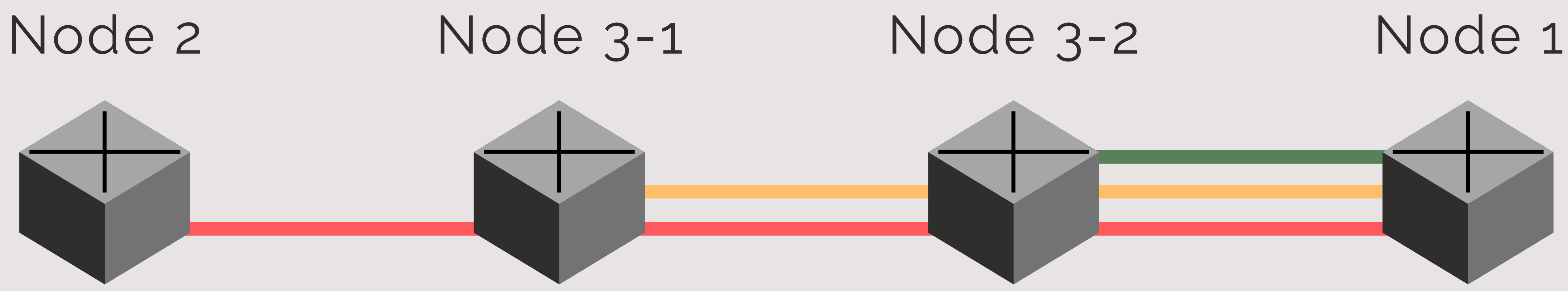
# MLP

TRAINING:
Based on the inputs, retrieve the weights and threshold such that the cost function is at its minimum. Adjust the values through backwards propagation.

TESTING:
Feed the datum and activate the relative sigmoid neurons. Through feedforward process, reach the output.



# The Scenarios




### Scenario 0

This scenario is faultless

### Scenario 1-5

In this scenario there are either one or two faults: each one decreasing by -11dB the OSNR

### Scenario 6-8

In this scenario there is only one fault decreasing by a lot the OSNR

# The Dataset

| Scenario | Path/Groundtruth | | |
|---|---|---|---|
| 0 | 2-1/False | 31-1/False | 32-1/False |
| 1 | 2-1/True | 31-1/True | 32-1/False |
| 2 | 2-1/True | 31-1/True | 32-1/False |
| 3 | 2-1/True | 31-1/False | 32-1/False |
| 4 | 2-1/True | 31-1/True | 32-1/False |
| 5 | 2-1/True | 31-1/True | 32-1/False |
| 6 | 2-1/True | 31-1/False | 32-1/False |
| 7 | 2-1/True | 31-1/True | 32-1/True |
| 8 | 2-1/True | 31-1/True | 32-1/True |

## Groundtruth

Thanks to the split function, we have been able to identify the scenario of the file from its name.

After identifying the scenario, it has been appended the groundtruth to each line of the dataset.

## Whole dataset

All the lines of the files have been merged in a single dataset called DATASET

# The Normalization

## WHY THE NORMALIZATION IS SO IMPORTANT?

A database must be normalized to minimize redundancy and to ensure that only related data is stored in each table.
It also prevents any issues stemming from database modifications such as insertions, deletions, and updates.

```
normalized_dataset = (dataset - dataset.mean()) / dataset.std()
```

# Original

| | DATE | PK_WL[THZ] | LEVEL[dBm] | 3.0dB WD[nm] | CRT WL WL[NM] | 3.0dB PB[nm] | RIPPLE[dB] | CROSS TK[L] [dB] | CROSS TK[R] [dB] | OFFSET WL[nm] | OFFSET LEVEL[dBm] | NOISE[dBm] [NBW] | OSNR[dB] | File | GroundTruth |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **0** | 2019/10/09 08:18:43 250 | 194.801 | -21.022 | 181009301 | 194801720 | 381133933.0 | 13.703 | 12.889 | 12.864 | 0.0 | 0.0 | -41.605 | 20.527 | 2 | False |
| **1** | 2019/10/09 08:18:44 290 | 194.801 | -21.016 | 179616980 | 194801742 | 381766910.0 | 13.774 | 12.820 | 12.802 | 0.0 | 0.0 | -41.604 | 20.533 | 2 | False |
| **2** | 2019/10/09 08:18:45 301 | 194.801 | -21.016 | 180376671 | 194801862 | 376577182.0 | 13.497 | 13.093 | 13.083 | 0.0 | 0.0 | -41.620 | 20.549 | 2 | False |
| **3** | 2019/10/09 08:18:46 285 | 194.801 | -21.016 | 180376671 | 194801862 | 376577182.0 | 13.497 | 13.093 | 13.083 | 0.0 | 0.0 | -41.725 | 20.502 | 2 | False |
| **4** | 2019/10/09 08:18:47 272 | 194.801 | -21.167 | 178731048 | 194801818 | 385058355.0 | 13.972 | 12.463 | 12.436 | 0.0 | 0.0 | -41.725 | 20.502 | 2 | False |

# Normalized

| | PK_WL[THZ] | LEVEL[dBm] | 3.0dB WD[nm] | CRT WL WL[NM] | 3.0dB PB[nm] | RIPPLE[dB] | CROSS TK[L][dB] | CROSS TK[R][dB] | NOISE[dBm][NBW] |
|---|---|---|---|---|---|---|---|---|---|
| **0** | 0.153127 | 0.508579 | -0.465391 | 0.299599 | 0.274882 | -0.372848 | 0.937200 | 0.932319 | 0.224355 |
| **1** | 0.153127 | 0.512984 | -0.523944 | 0.340002 | 0.280726 | -0.359917 | 0.919874 | 0.916736 | 0.224654 |
| **2** | 0.153127 | 0.512984 | -0.491996 | 0.560381 | 0.232812 | -0.410368 | 0.988422 | 0.987361 | 0.219878 |
| **3** | 0.153127 | 0.512984 | -0.491996 | 0.560381 | 0.232812 | -0.410368 | 0.988422 | 0.987361 | 0.188538 |
| **4** | 0.153127 | 0.402145 | -0.561201 | 0.479575 | 0.311115 | -0.323854 | 0.830235 | 0.824748 | 0.188538 |

# OSNR after normalization

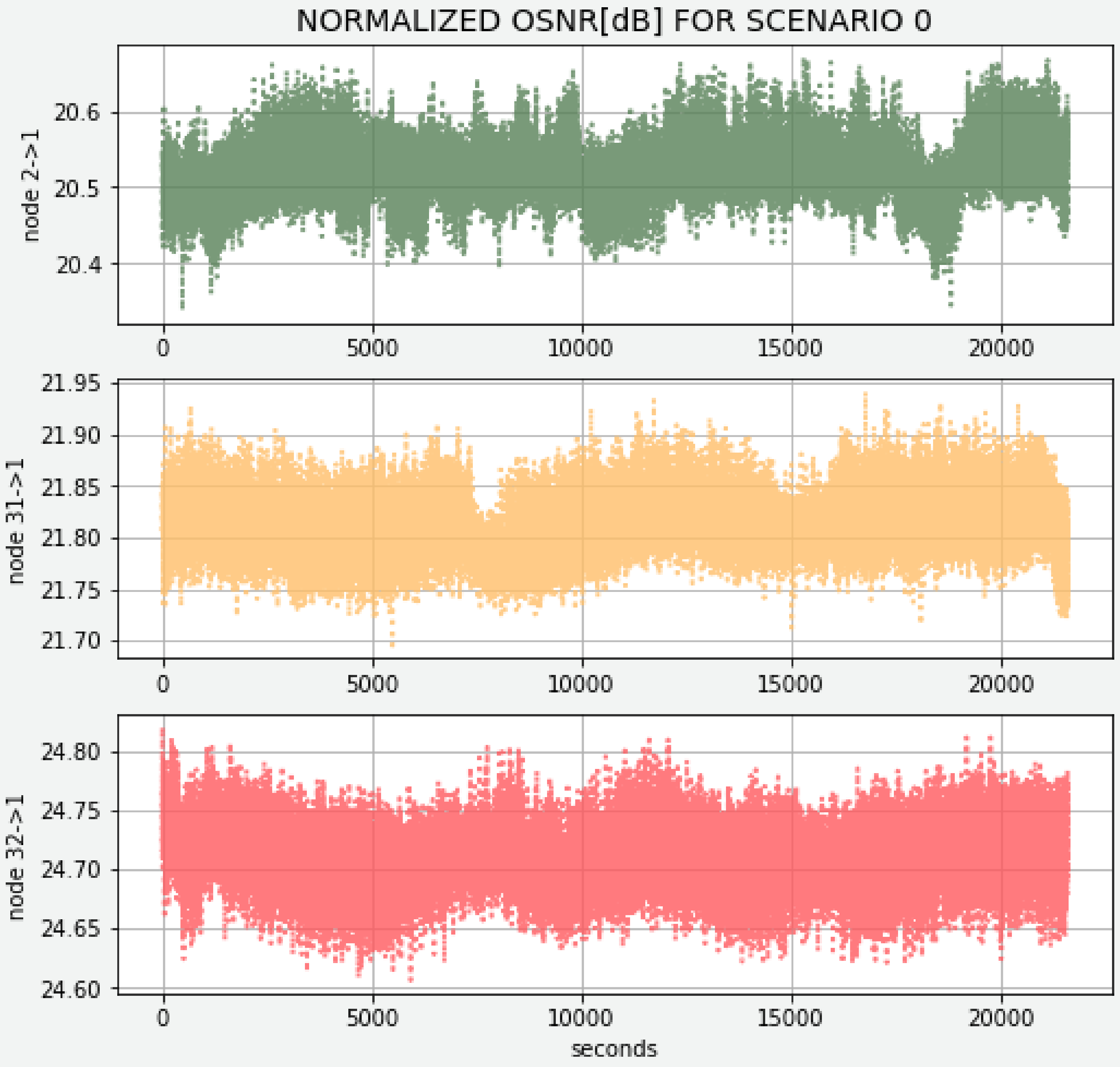
## FAULTLESS SCENARIO

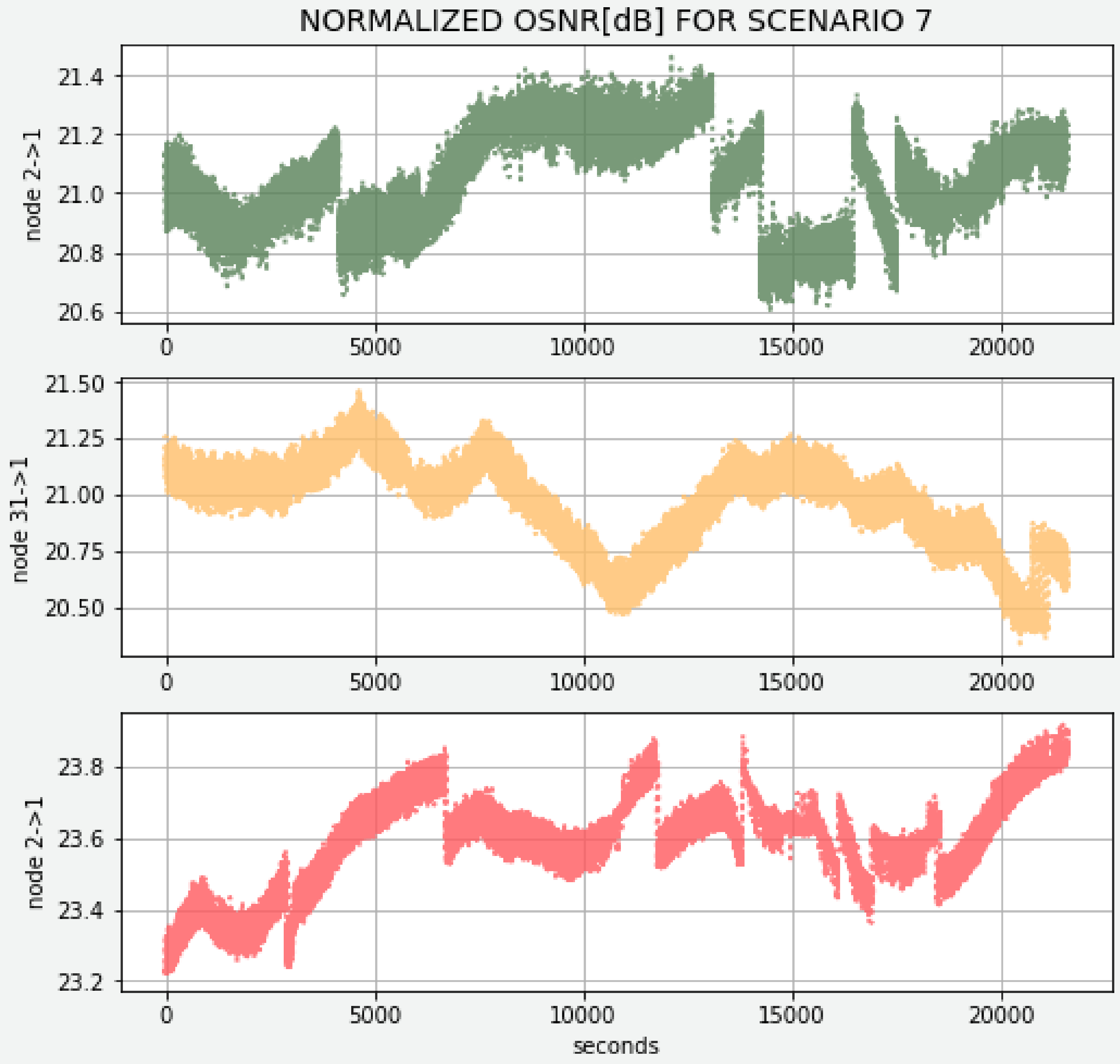
## FAULTY SCENARIO

NODE 2-1

NODE 31-2

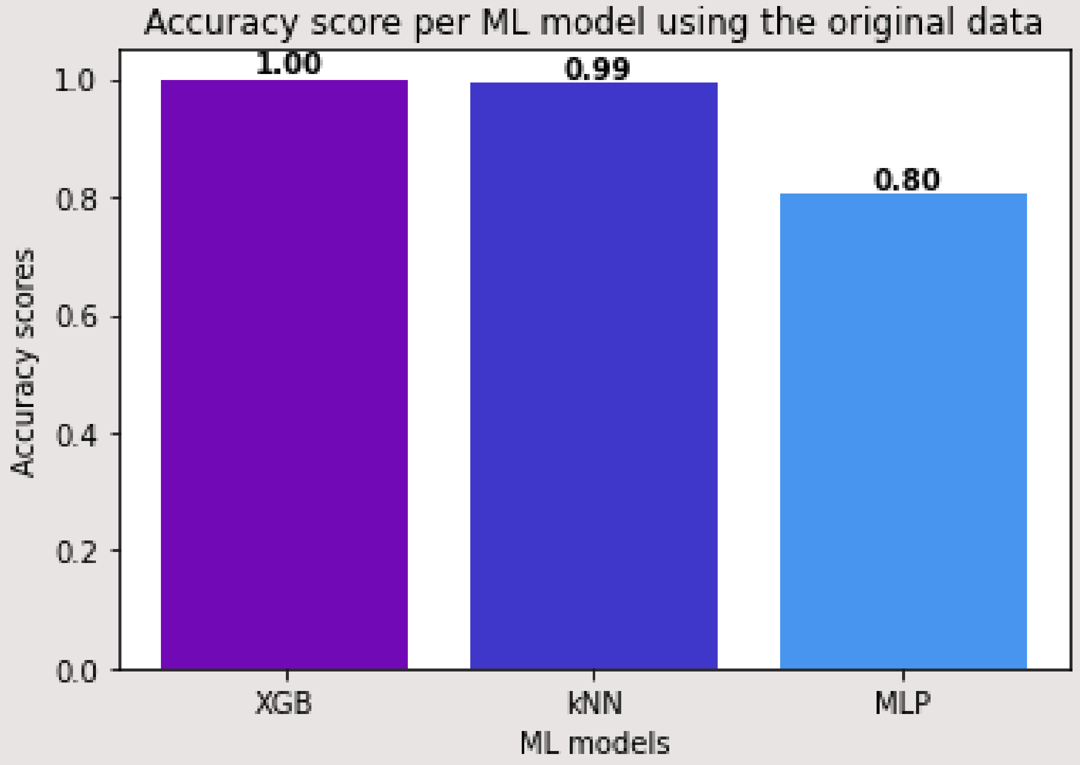
NODE 32-1



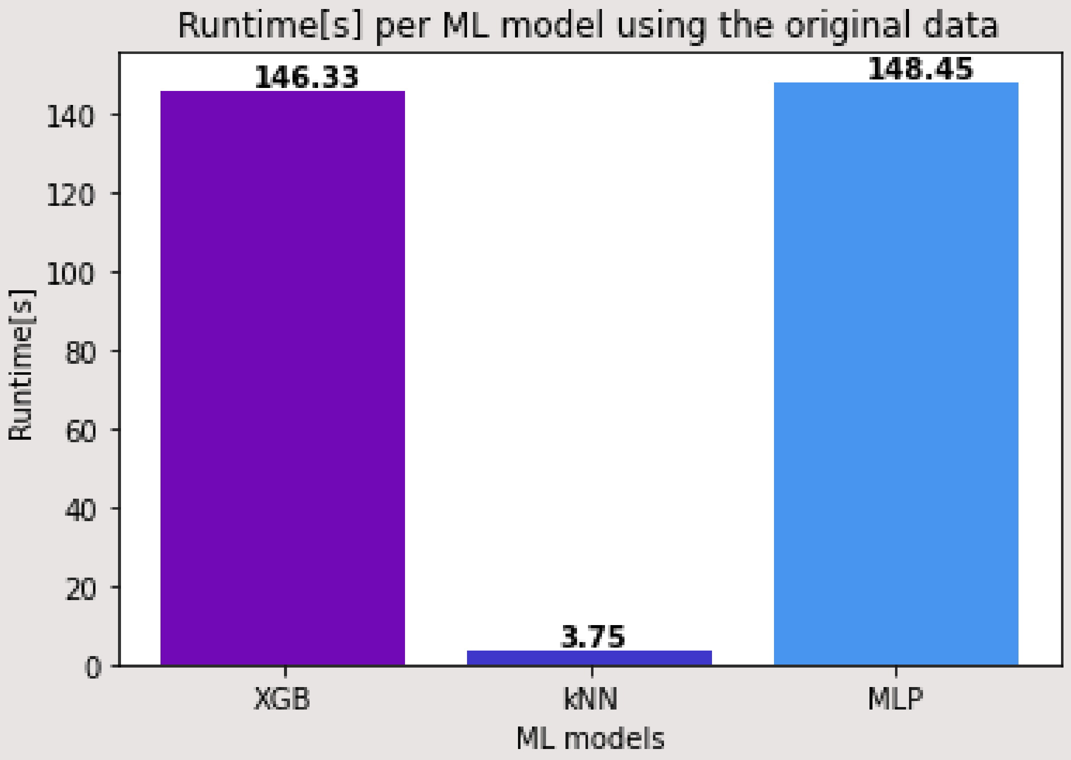



# Original Dataset

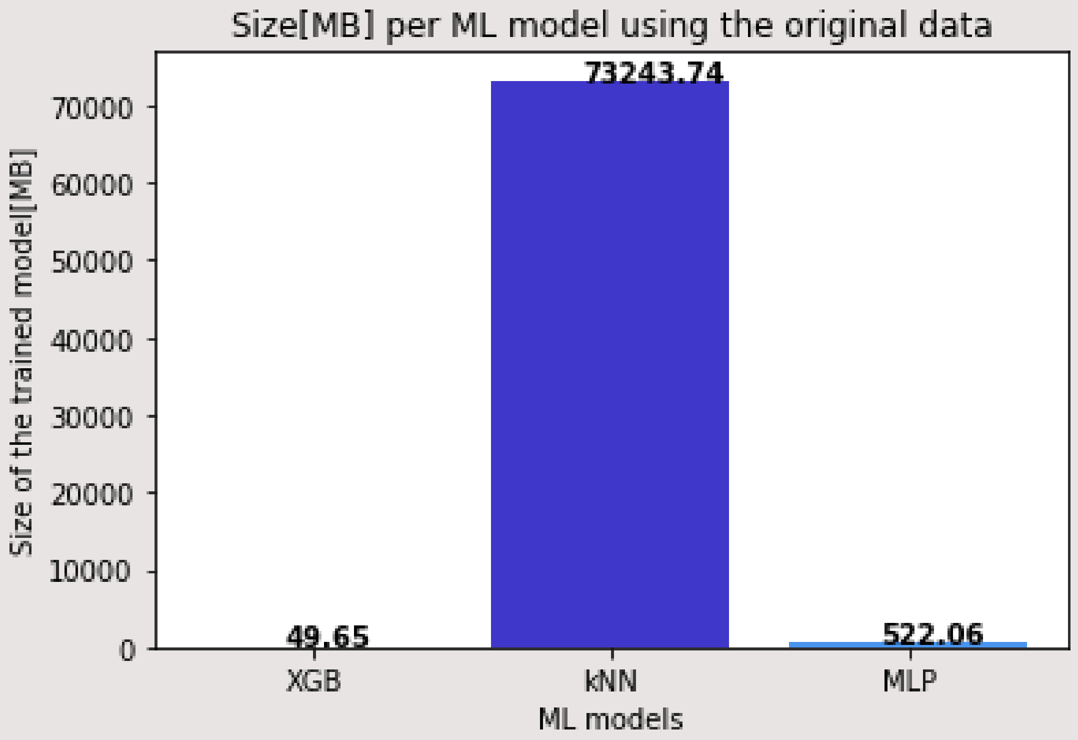
## ACCURACY




## PROCESSING TIME

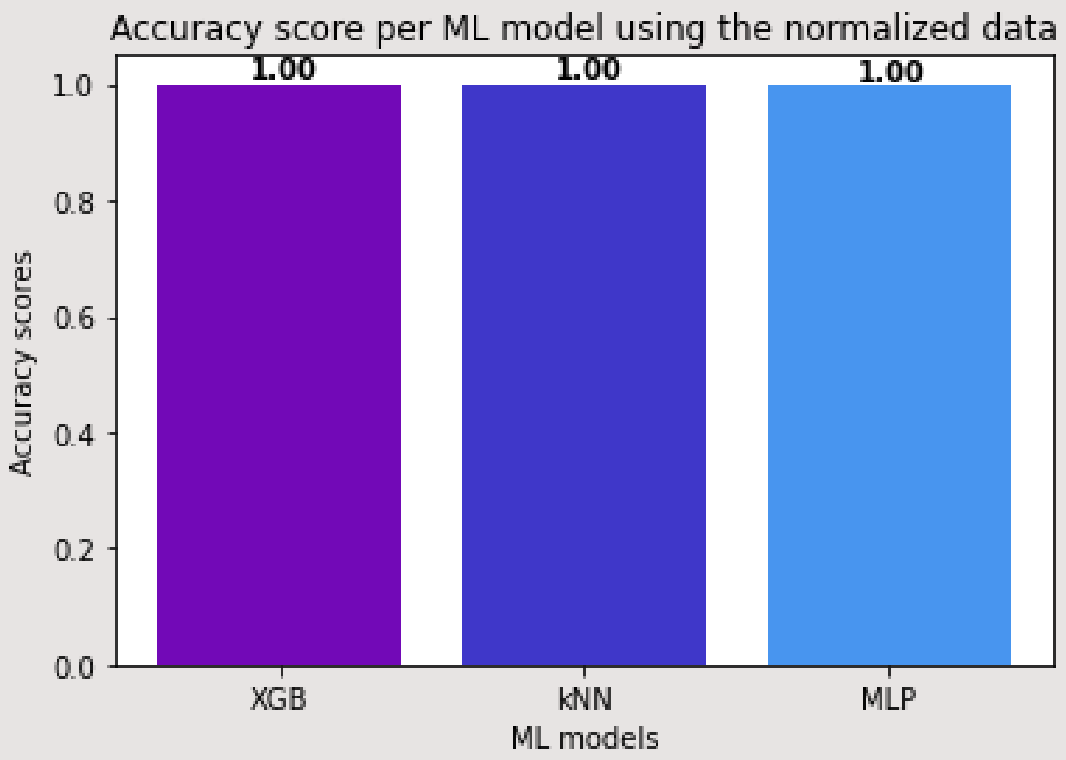



## SIZE OF THE MODEL

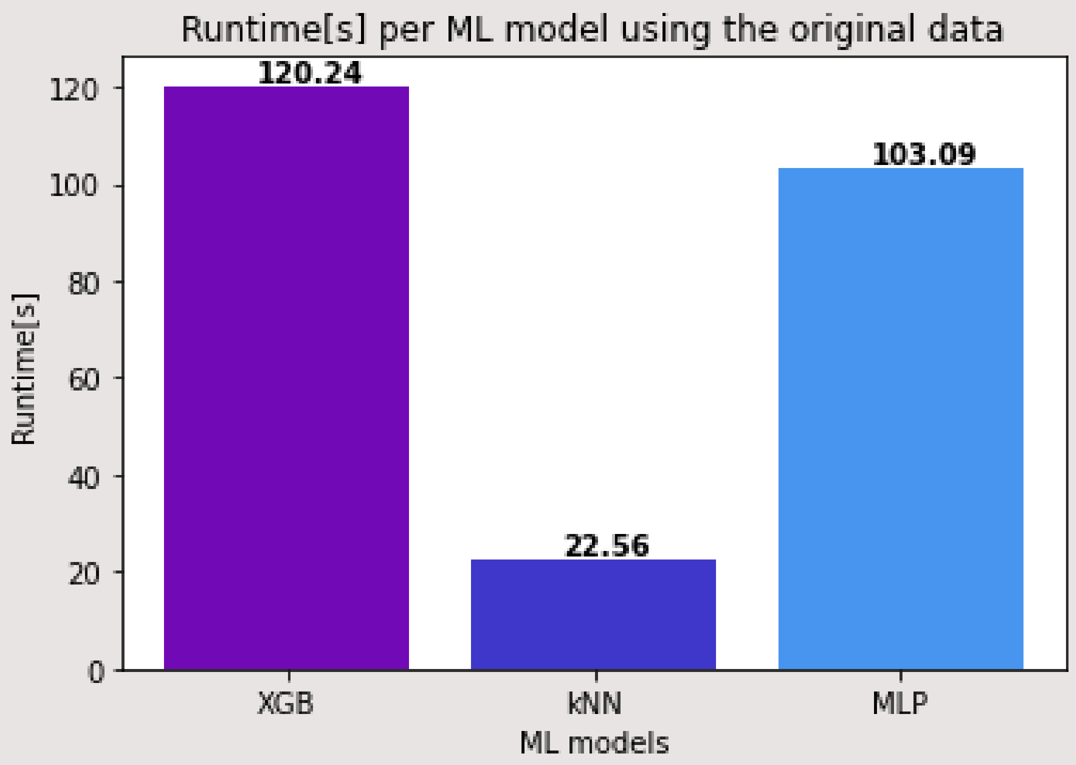


# Normalized Dataset

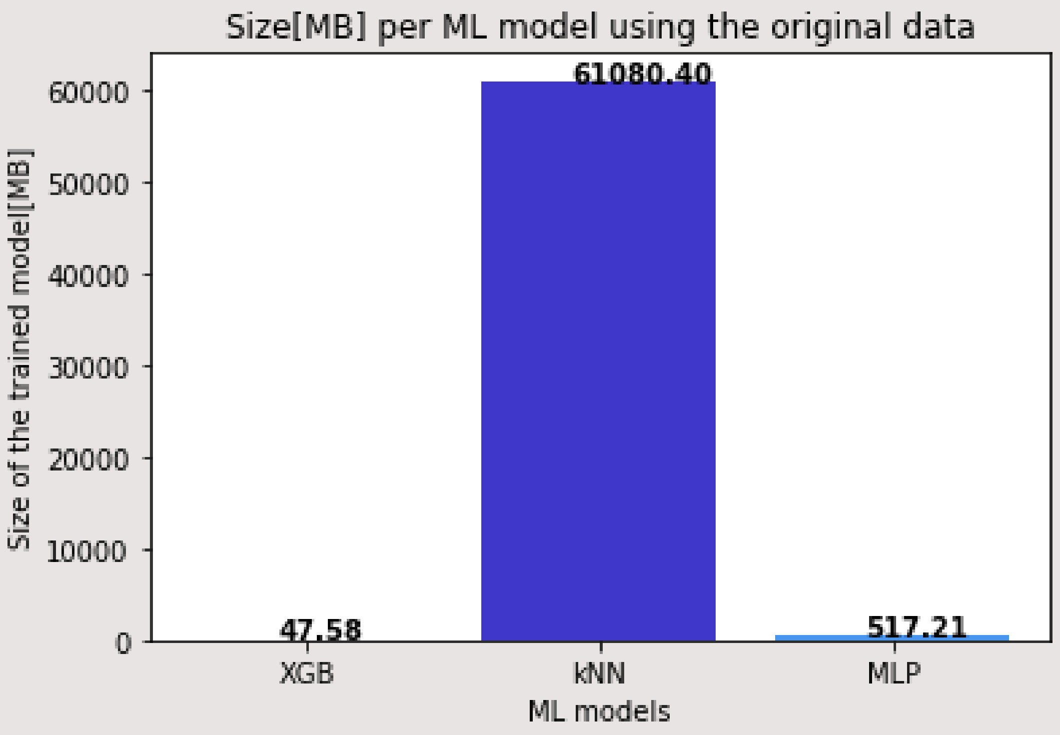
## ACCURACY




## PROCESSING TIME



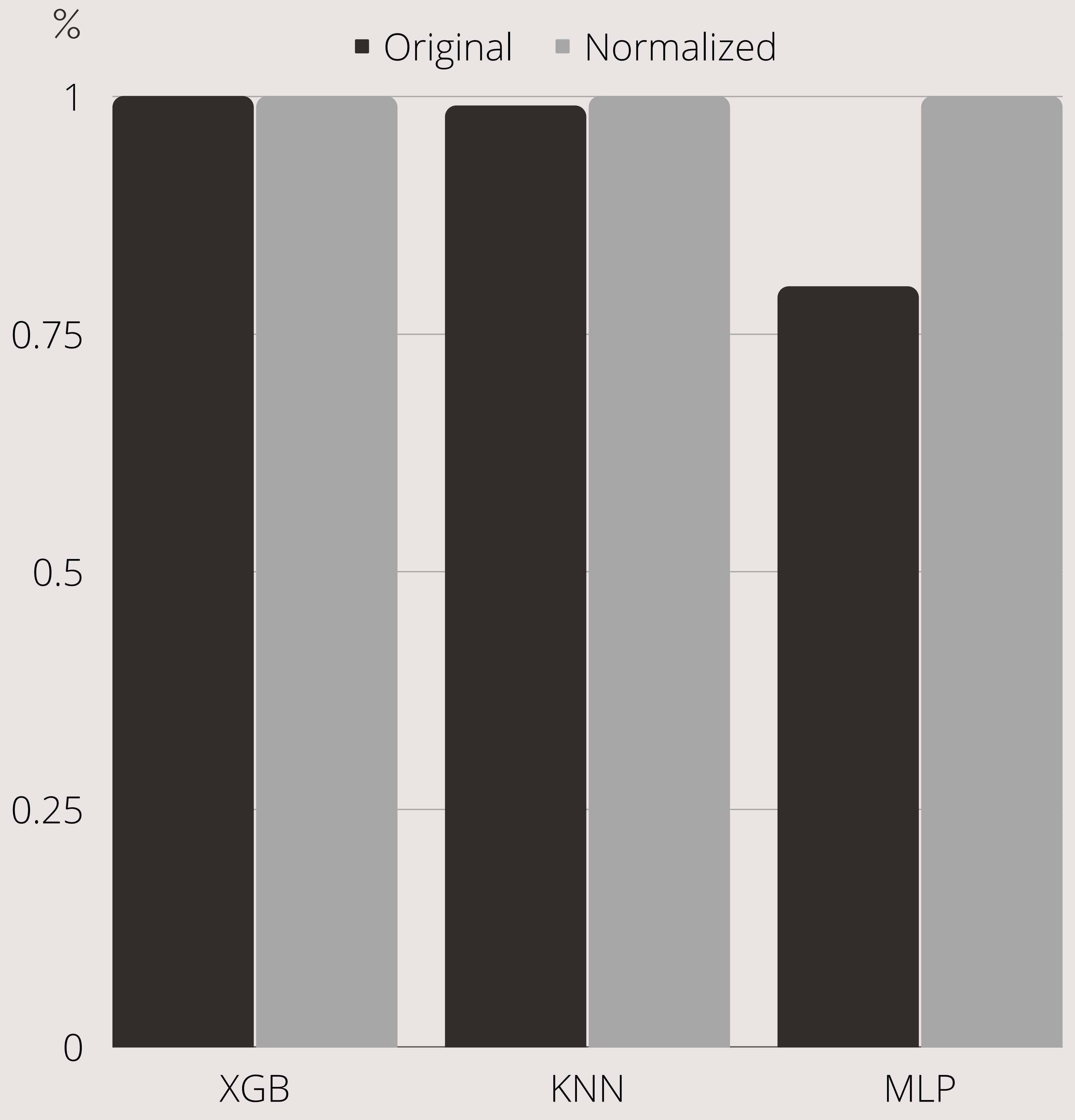

## SIZE OF THE MODEL



# Comparison

ACCURACY

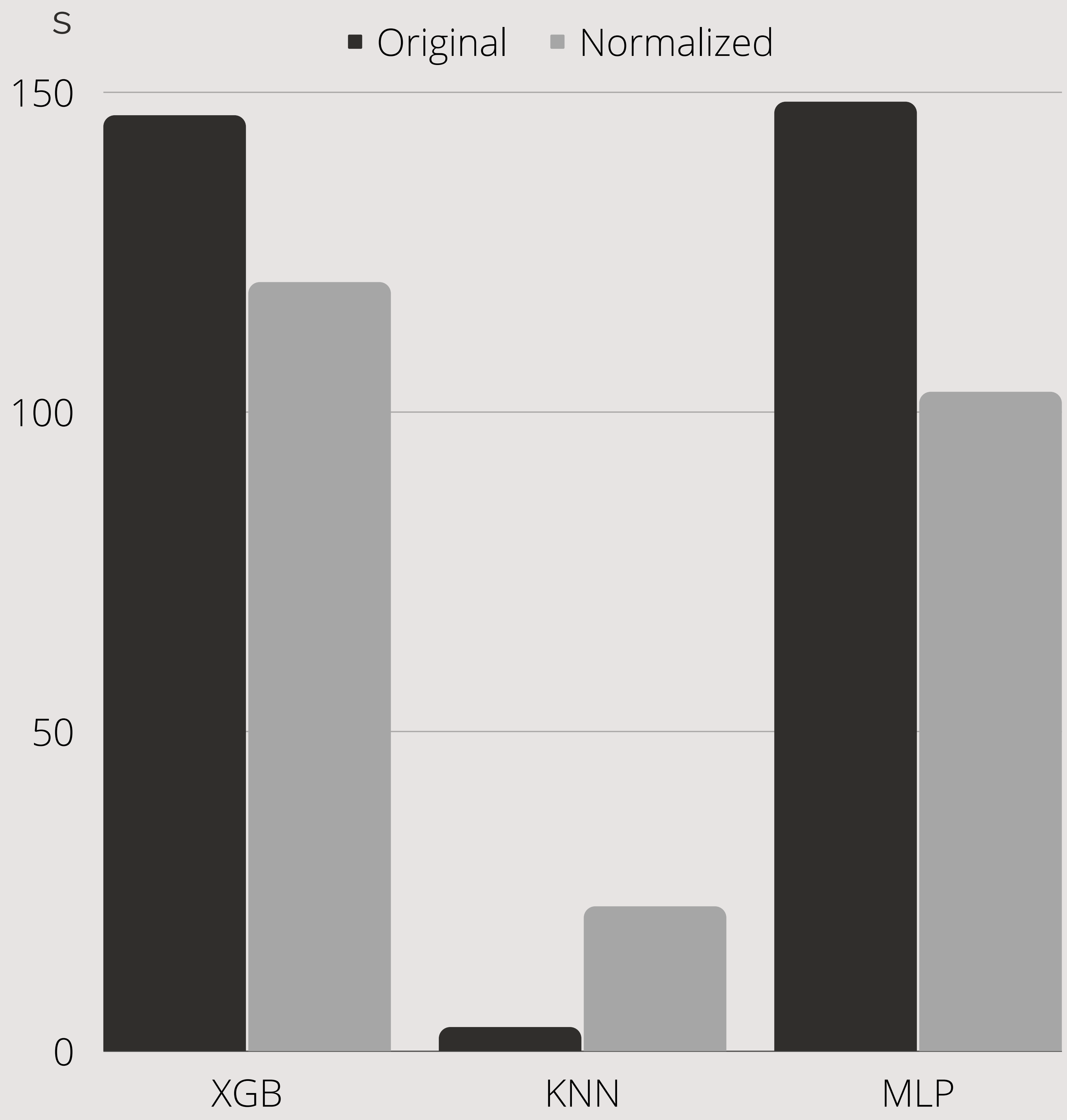
As expected, feeding a normalized dataset into the classifiers increase the accuracy score



# Comparison

## PROCESSING TIME

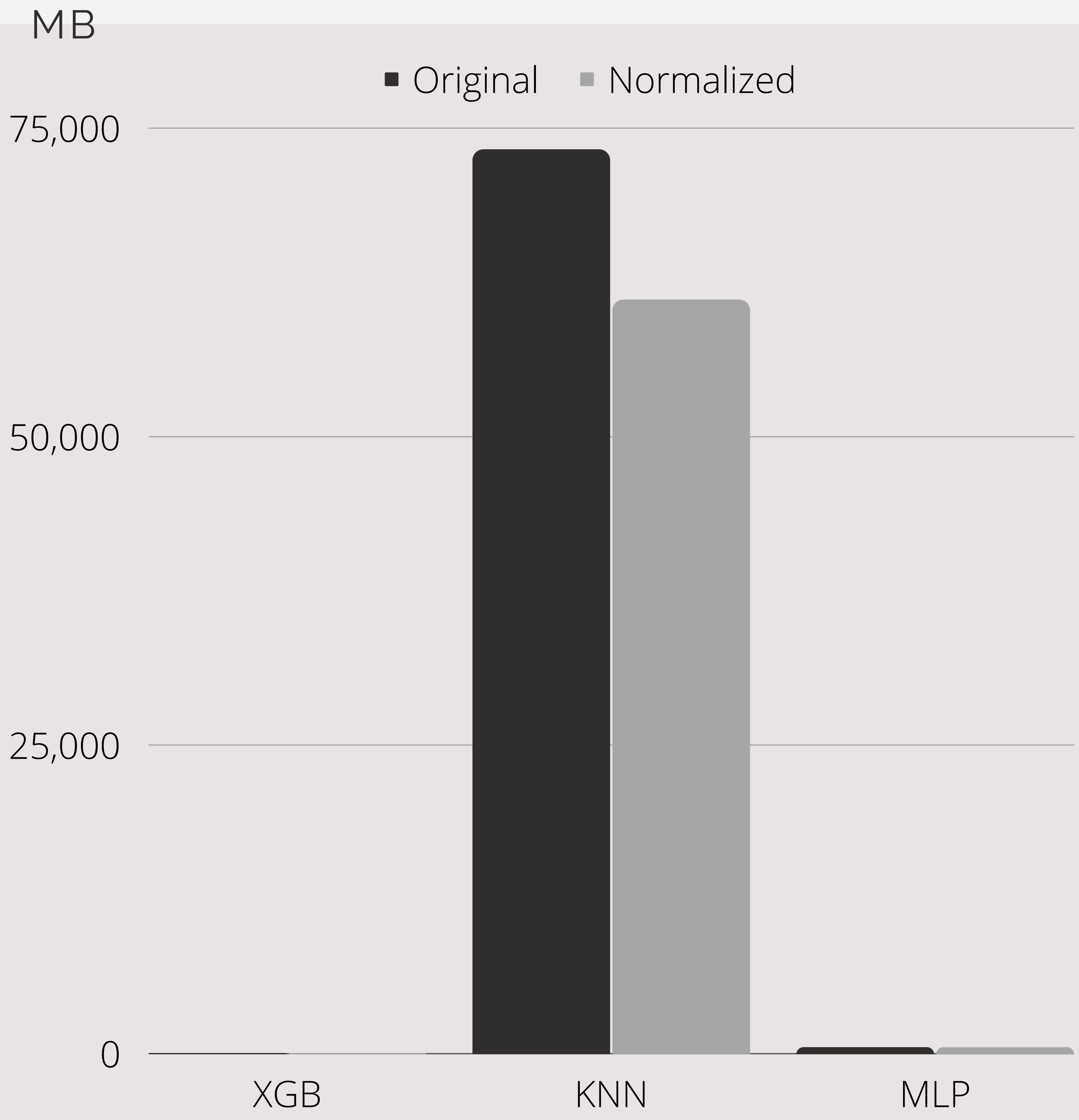
It can be observed that for the XGB and MLP classifiers, the use of a normalized dataset impacts the processing time with a decrease of almost 20% of it.
The KNN is the only algorithm in which the non-normalized dataset runtime is less than the normalized ones.

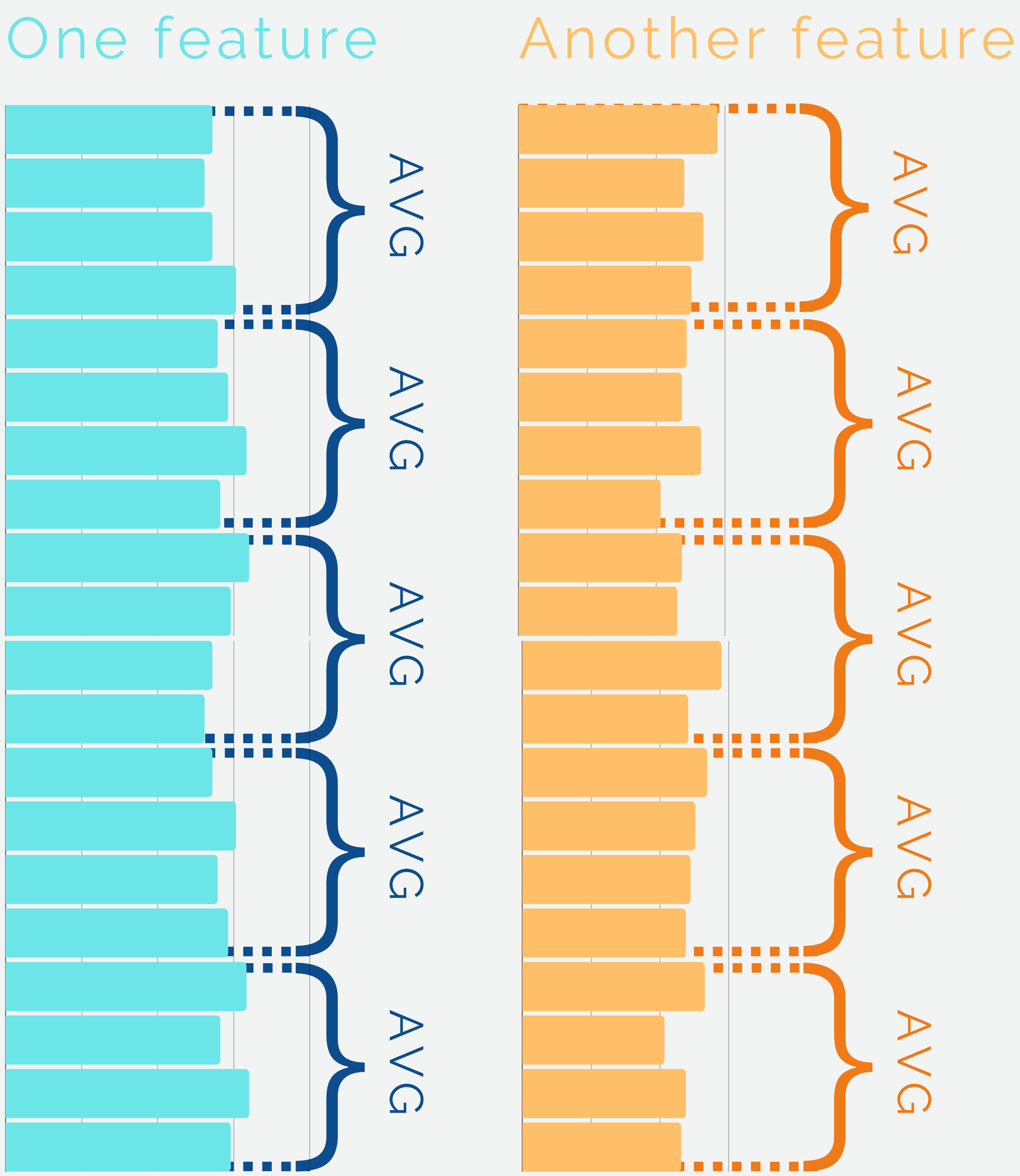


# Comparison

## SIZE OF THE MODEL

As the runtime, the size of the model is an important parameter to take into account when testing ML algorithms. In all the scenarios, the use of a normalized dataset allows to obtain higher accuracy score with a smaller size of the model.






In the picture: W=4, S=0

# Resampling

## Sampling Width

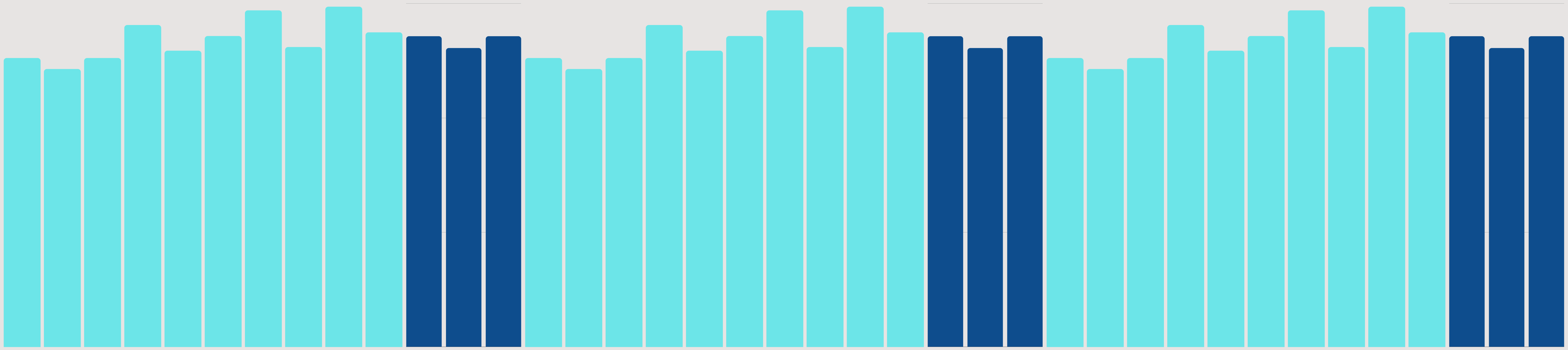
Let's call the Sampling width "W"
Every group of W rows,
we compute the average per feature.

## Size of dataset

If the original dataset had N rows
The resulting dataset will have $\lfloor N/W \rfloor$ rows

If we consider using a spacing equal to S,
The resulting dataset will have $\lfloor (N-W+S)/W \rfloor$ rows

# Changing Window duration



Window = 10

Spacing = 13

In the picture:
spacing is bigger than windows

Keeping the spacing constant:
- The higher the windowlength
- The higher is the correlation between windows
- The smaller the dataset

Window < Spacing
Window = Spacing
Window > Spacing

# Changing W

## KNN

Accuracy decreases whenever the Spacing is too small with respect to the window length. For KNN it's hard to classify samples which are highly correlated

## XGBC

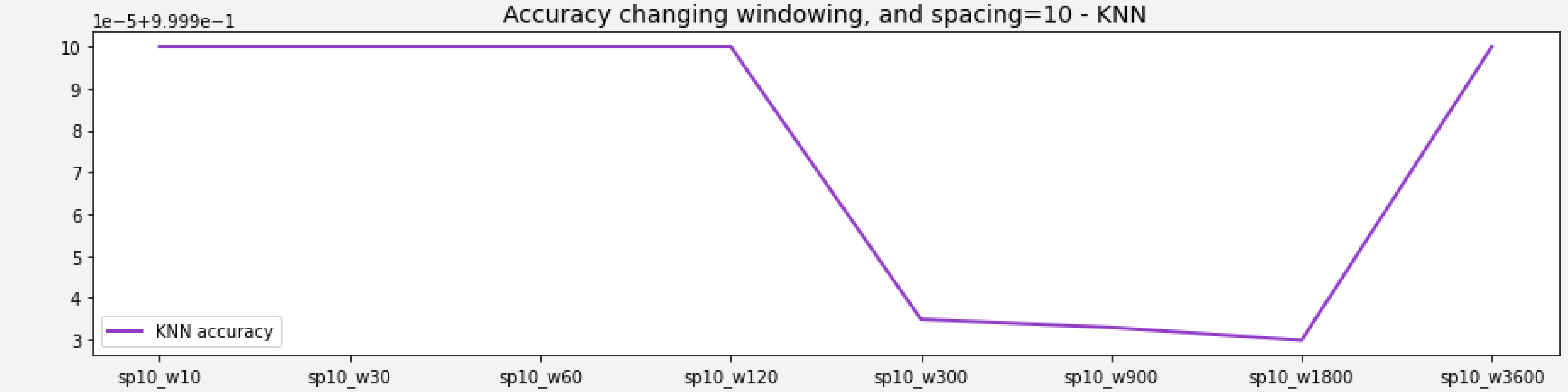
Accuracy tends to be higher for smaller SWR (spacing window ratio)

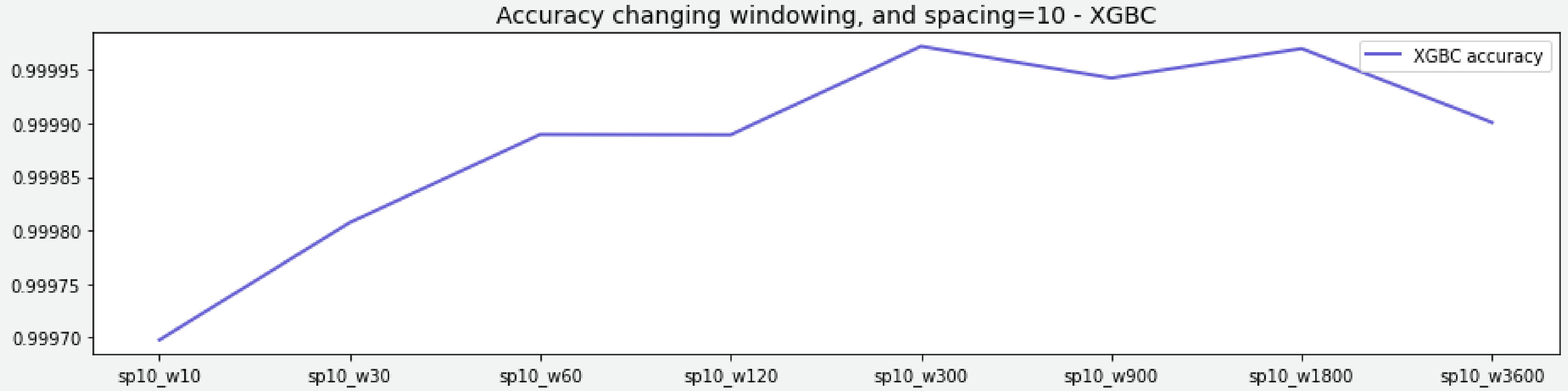
## MLP

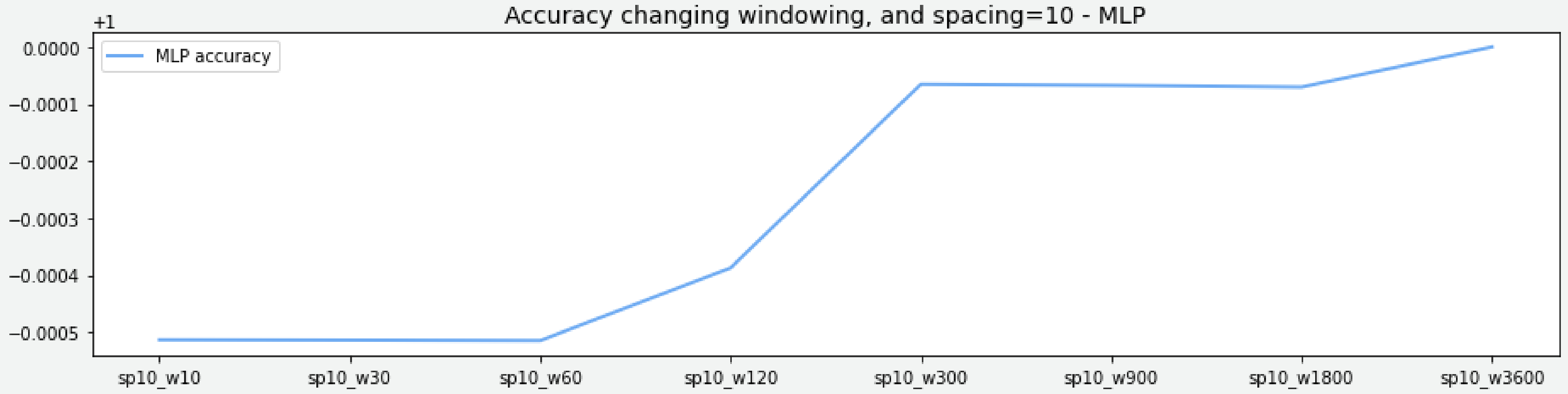
Accuracy increases drastically for SWR > 1/30, and from there it remains constant

# S = 10

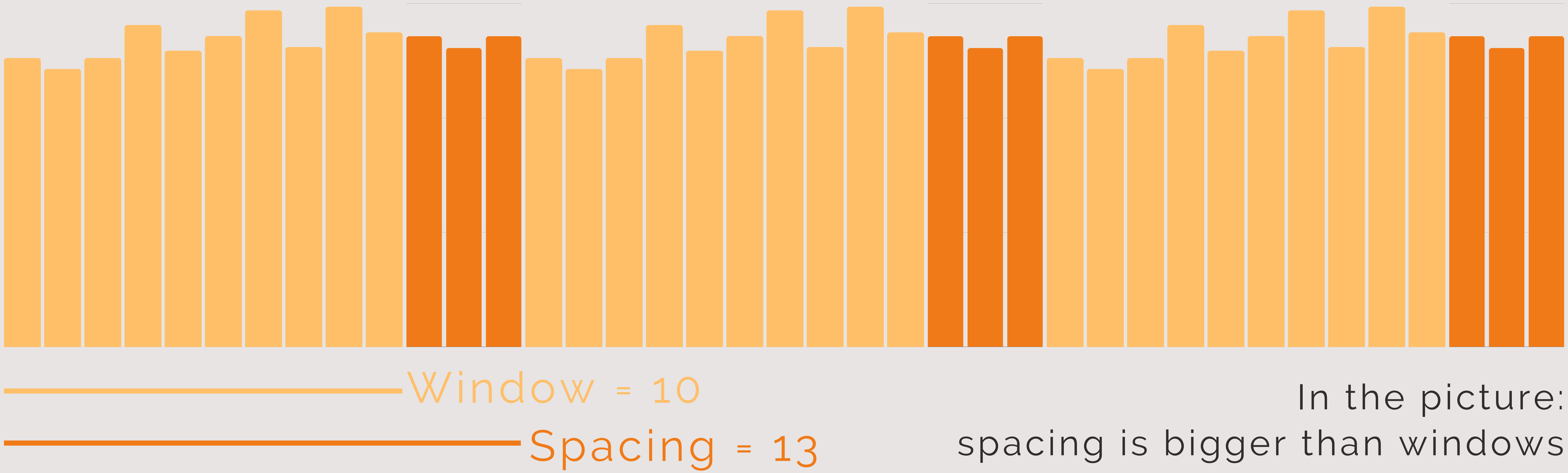
Spacing is smaller than window length



# Changing Spacing




In the picture:
spacing is bigger than windows

Keeping the windowlength constant:
- The higher the spacing
- The smaller the overlapping between windows
- The smaller the dataset

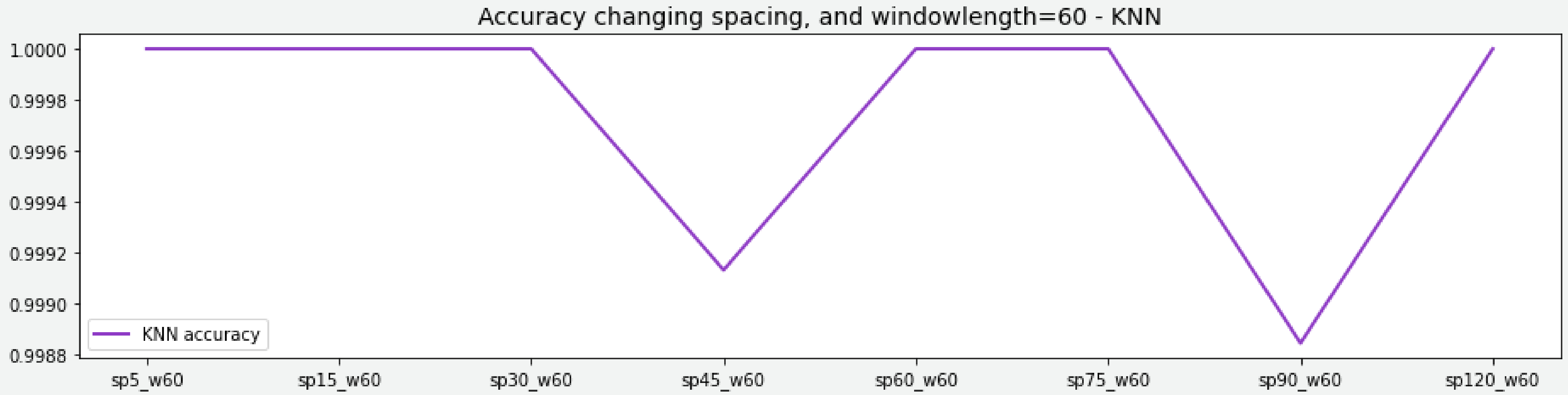
Window < Spacing
Window = Spacing
Window > Spacing

# Changing S

# W = 60

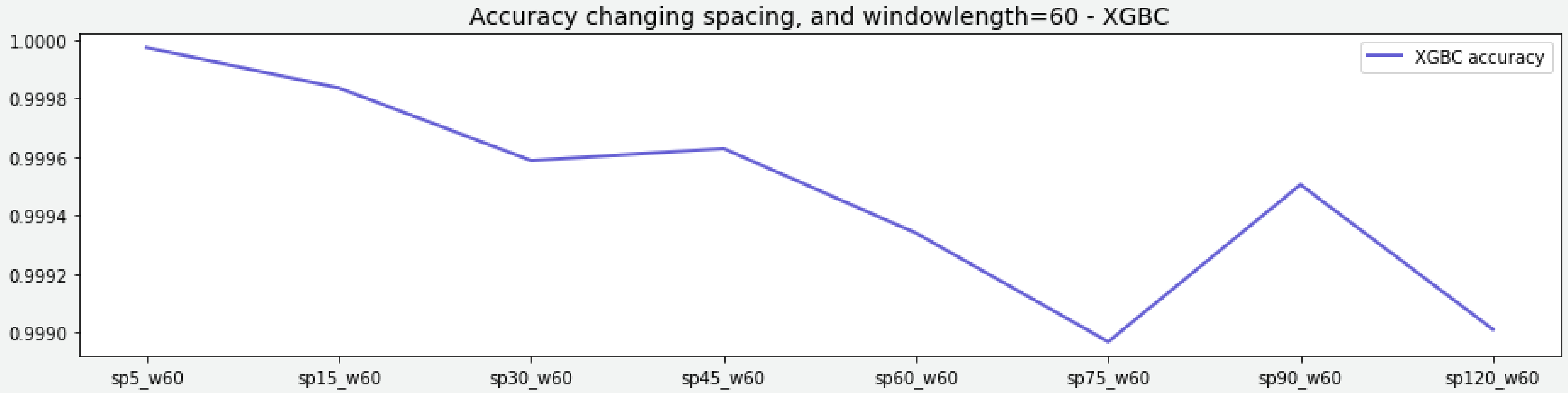
## KNN

Using a spacing of 45s and 90s decrease the accuracy of the KNN classifier leading to samples less correlated each other.



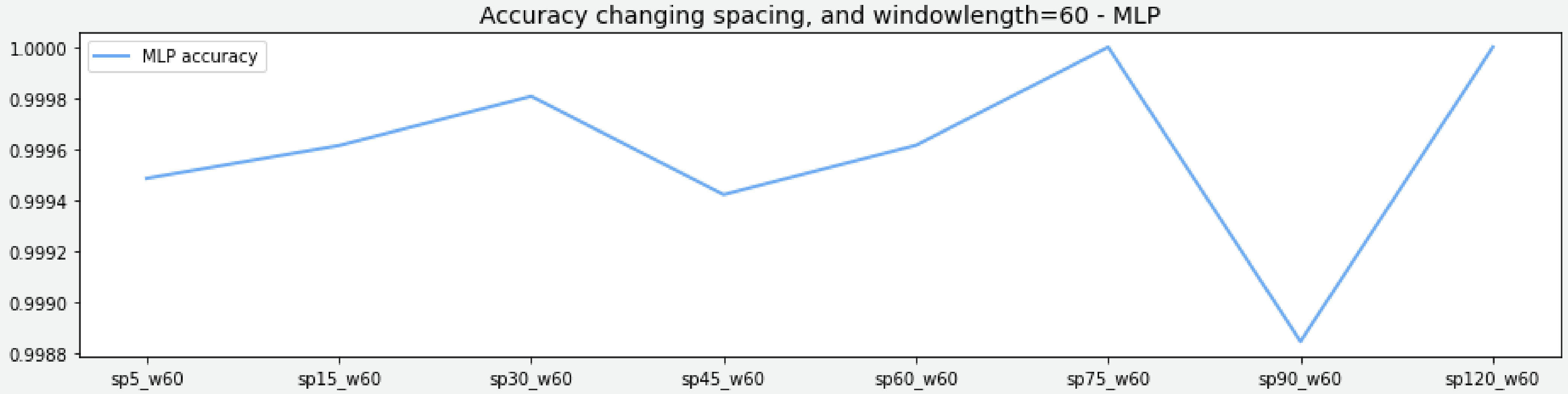

## XGBC

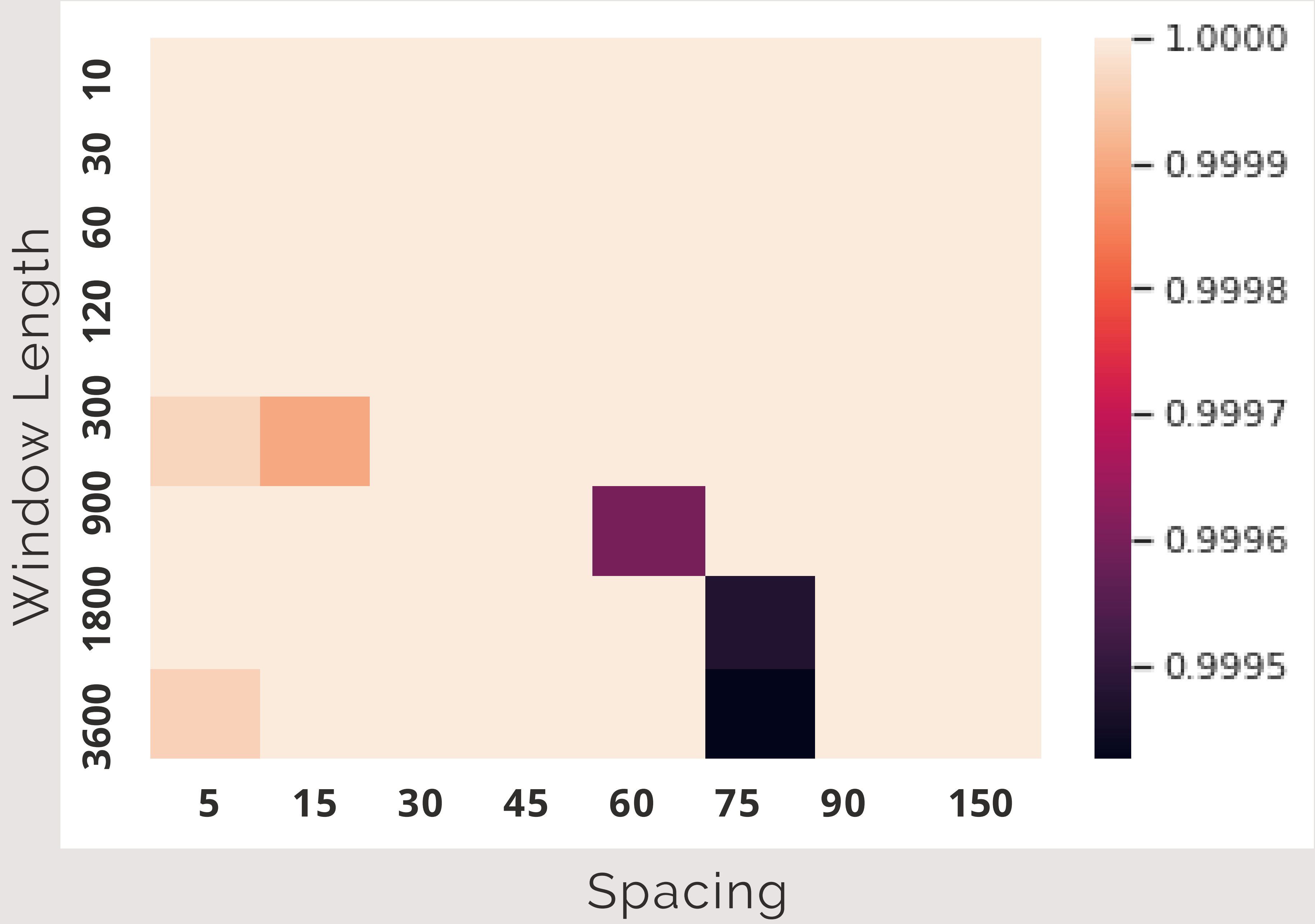
The accuracy tends to  decrease with the increasing of the Spacing




## MLP

The accuracy has a flactuating behaviour with the increase of the Spacing



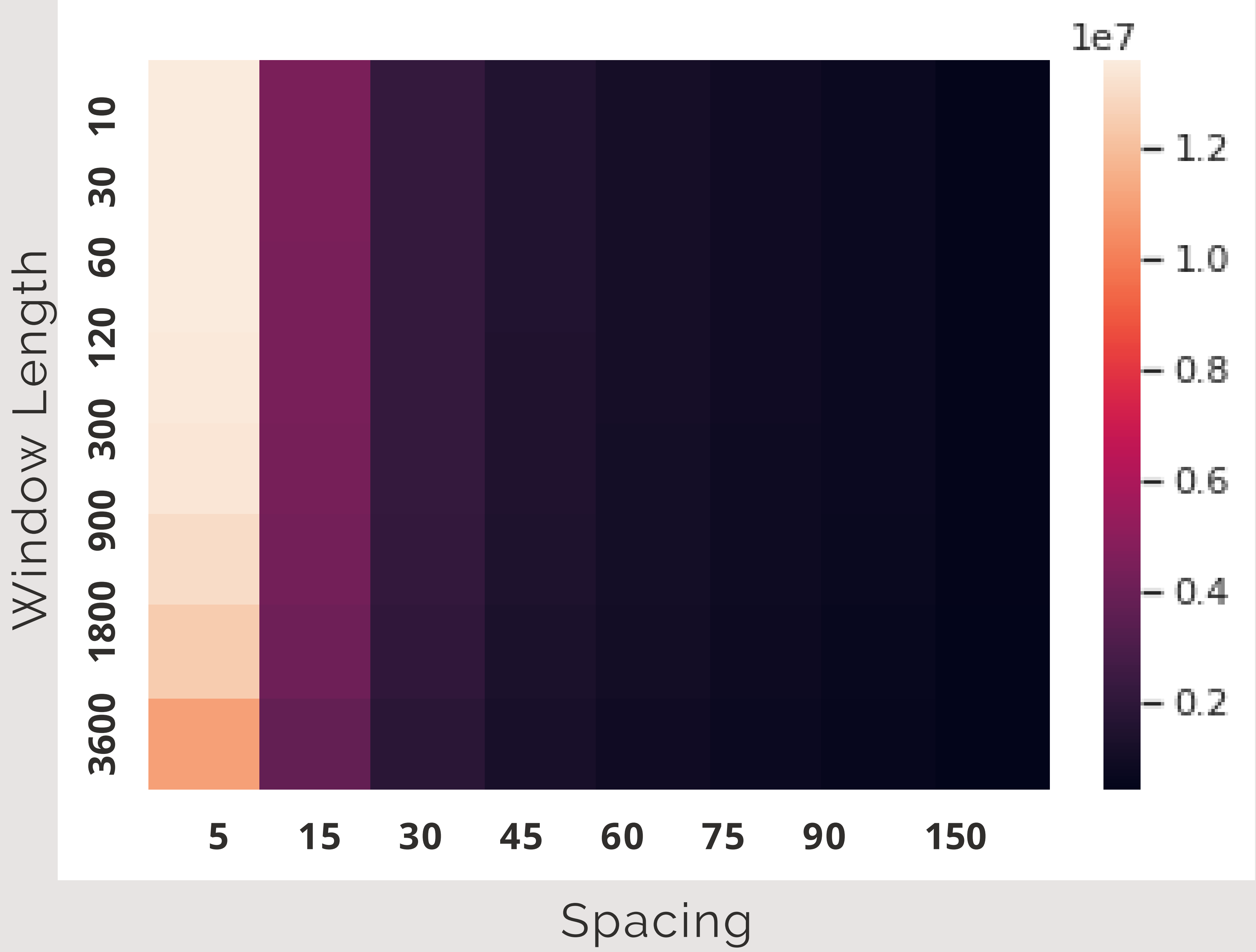



# KNN

## ACCURACY

Accuracy is extremely high,
no matter what's the windowlength or the spacing.

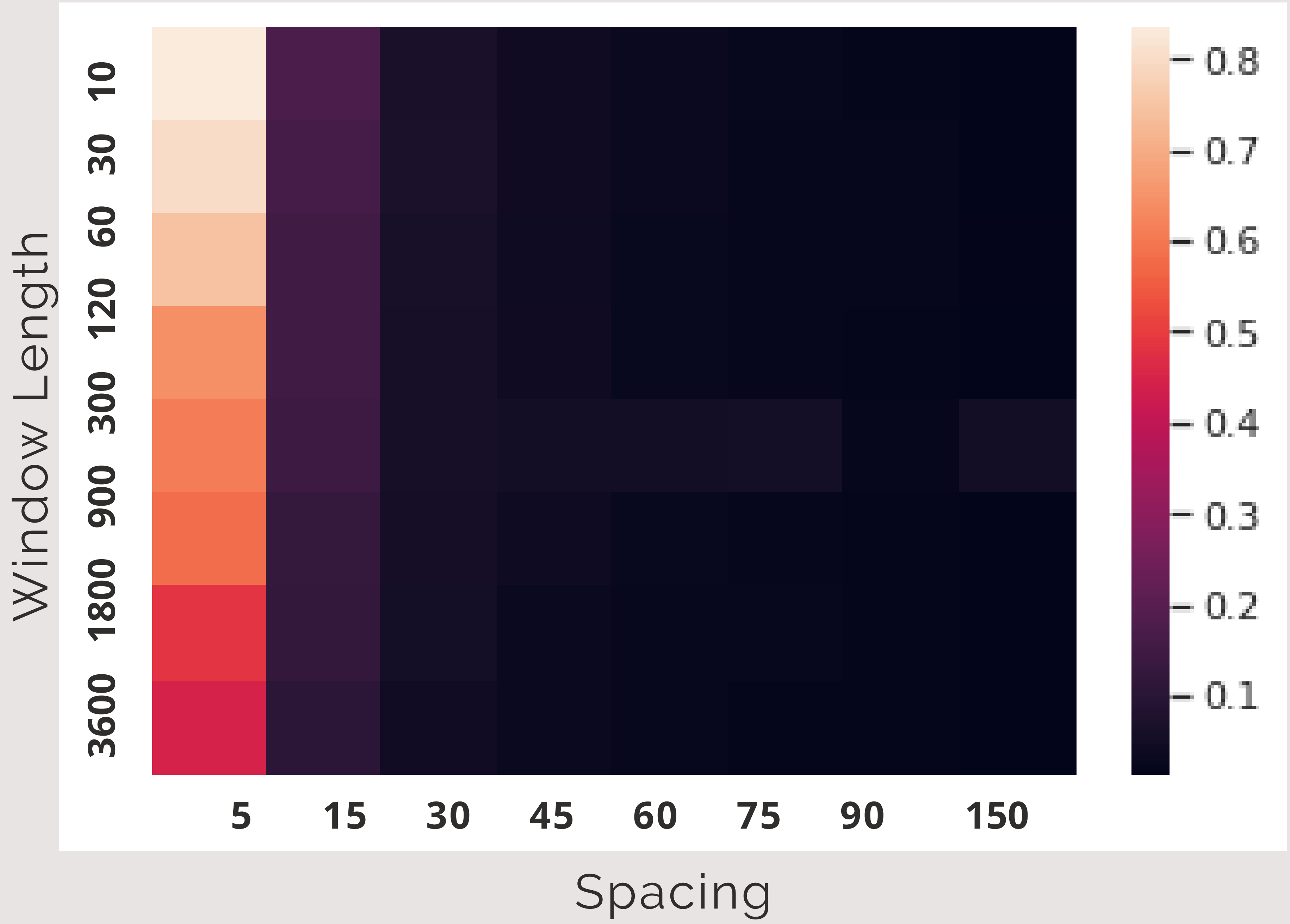
In any case the dataset is complete enough to ensure high accuracy




# KNN

## SIZE OF THE MODEL

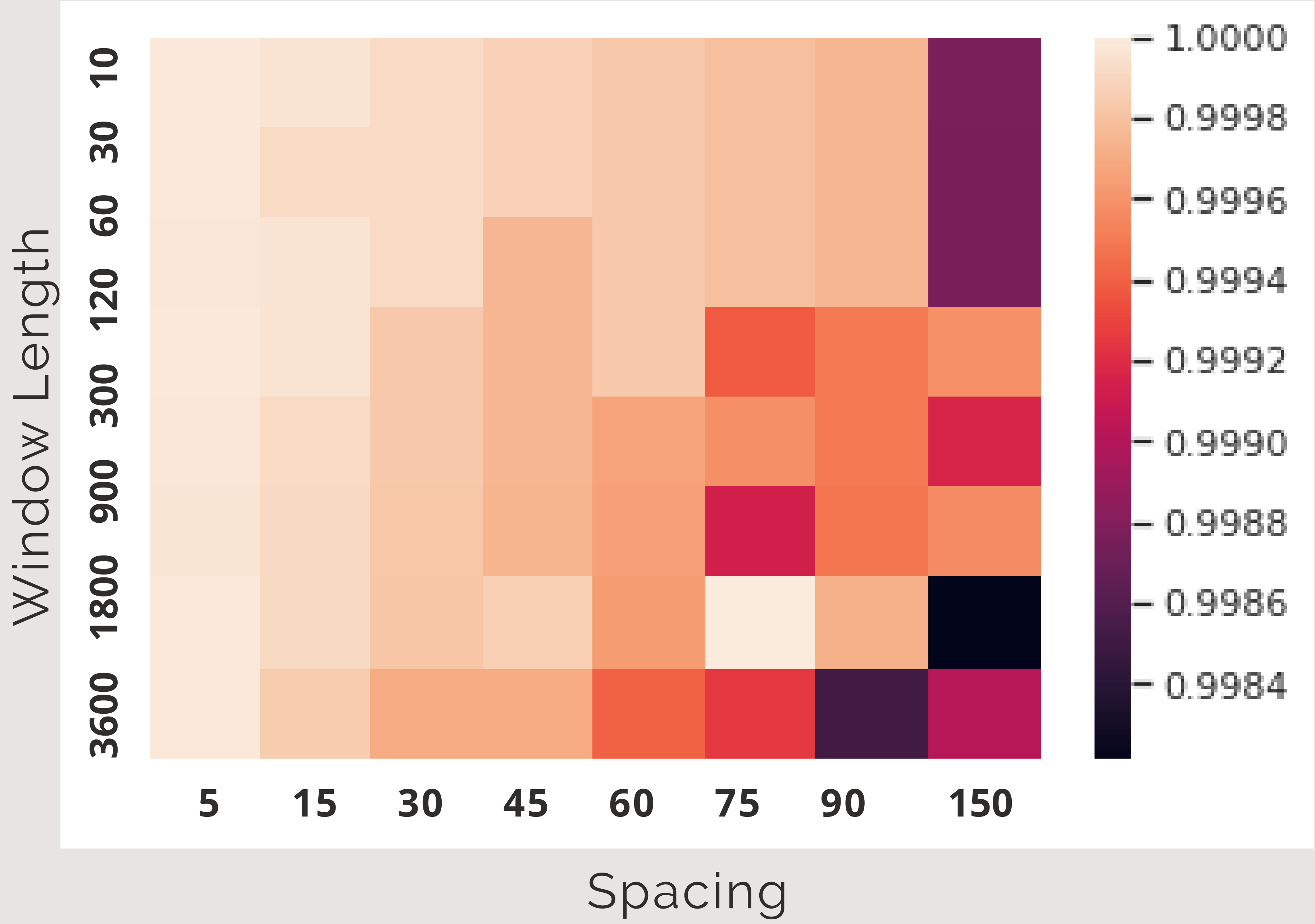
As mentioned before,
increasing the spacing causes the dataset to decrease in size,
and as a result the model as well will be extremely smaller with respect to the case spacing = 1.




# KNN

## PROCESSING TIME

As the spacing increases,
the number of rows in the dataset decreases, therefore the processing time decreases accordingly.

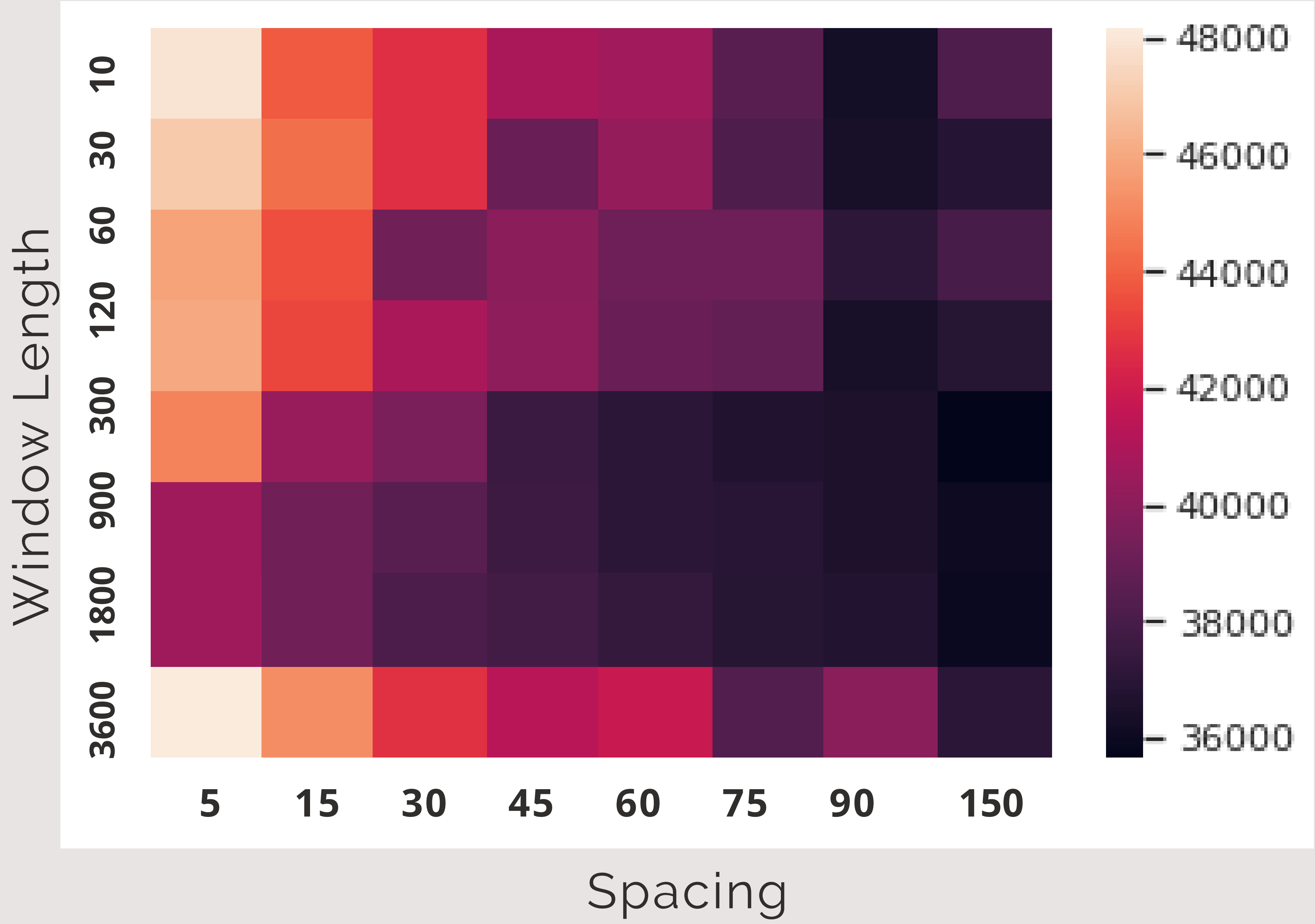



# XGBC

## ACCURACY

Increasing the spacing,
the correlation between rows decreases.
As a result, it's harder to label correctly a sample through XGBC algorithm.

N.B. the lowest accuracy obtained with XGBC is higher than the lowest accuracy obtained with KNN.

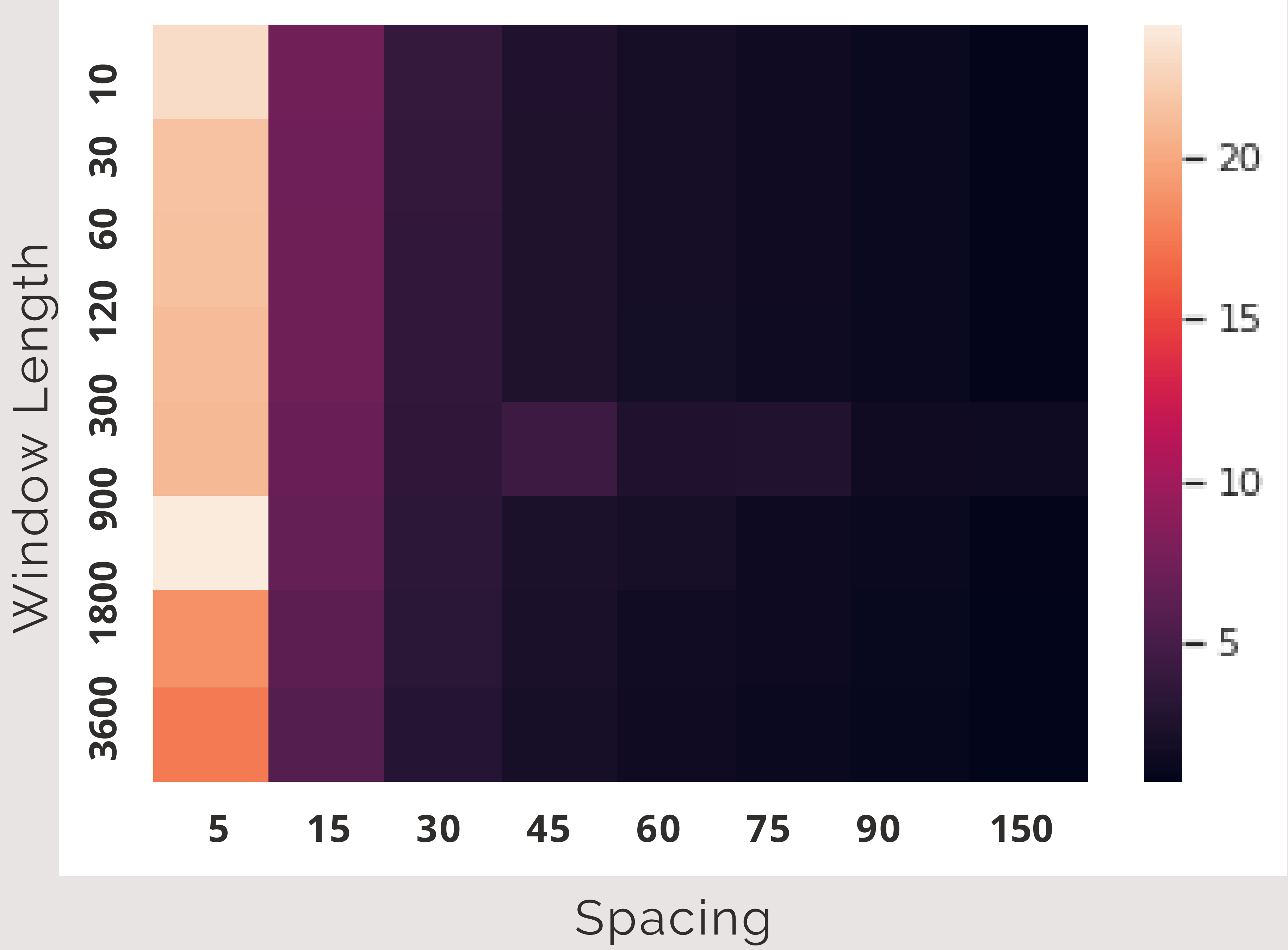



# XGBC

## SIZE OF THE MODEL

With the increase of the spacing the size of the model decreases.
With the increase of the windowlength the size of the model decreases.

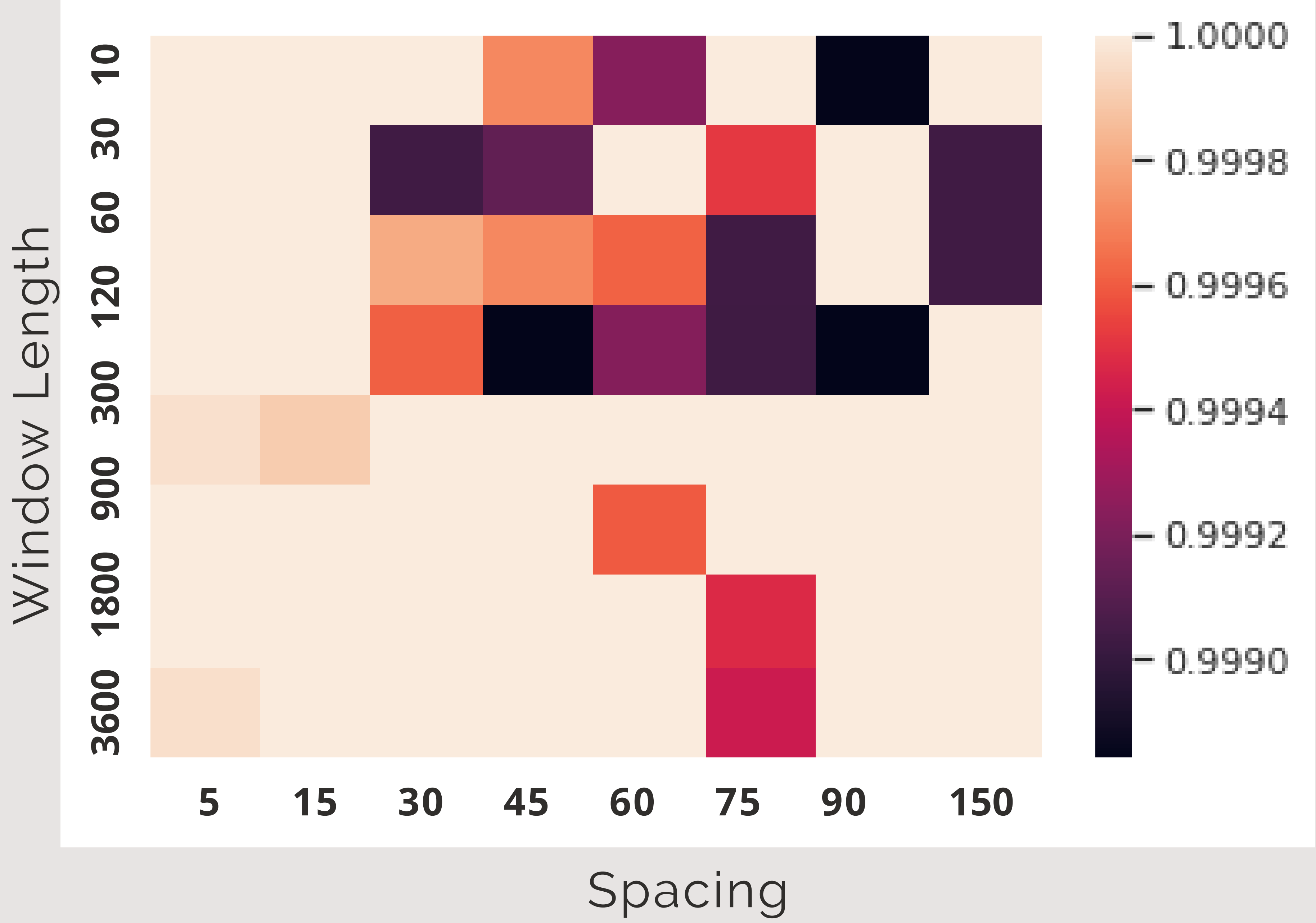
In both cases: this is due to the dataset having a smaller number of rows




# XGBC

## PROCESSING TIME

Just like in KNN algorithm:
As the spacing increases,
the number of rows in the dataset
decreases, therefore the processing
time decreases accordingly.

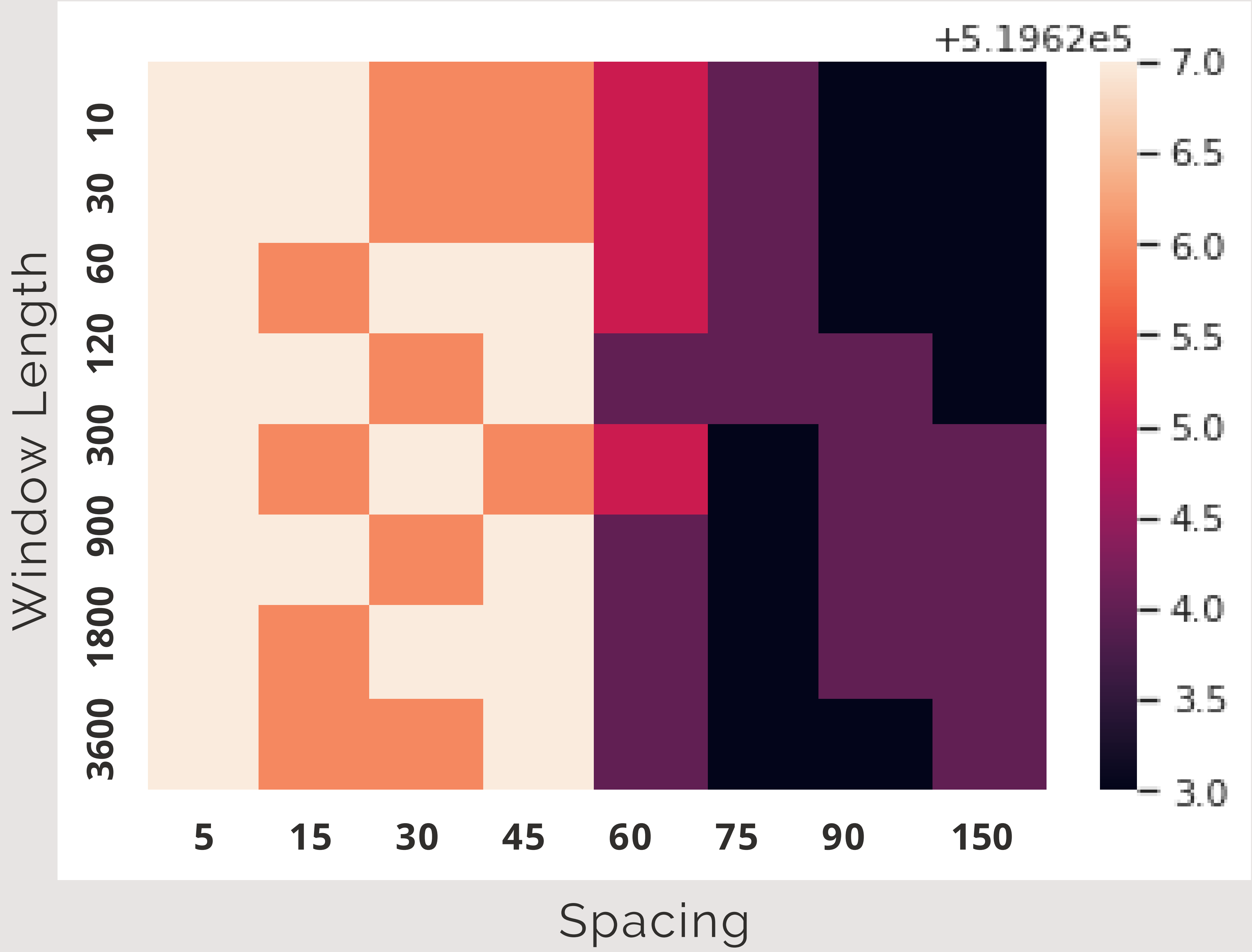



# MLP

## ACCURACY

Accuracy is extremely high,
no matter what's the windowlength or the spacing.

In any case the dataset is complete enough to ensure high accuracy

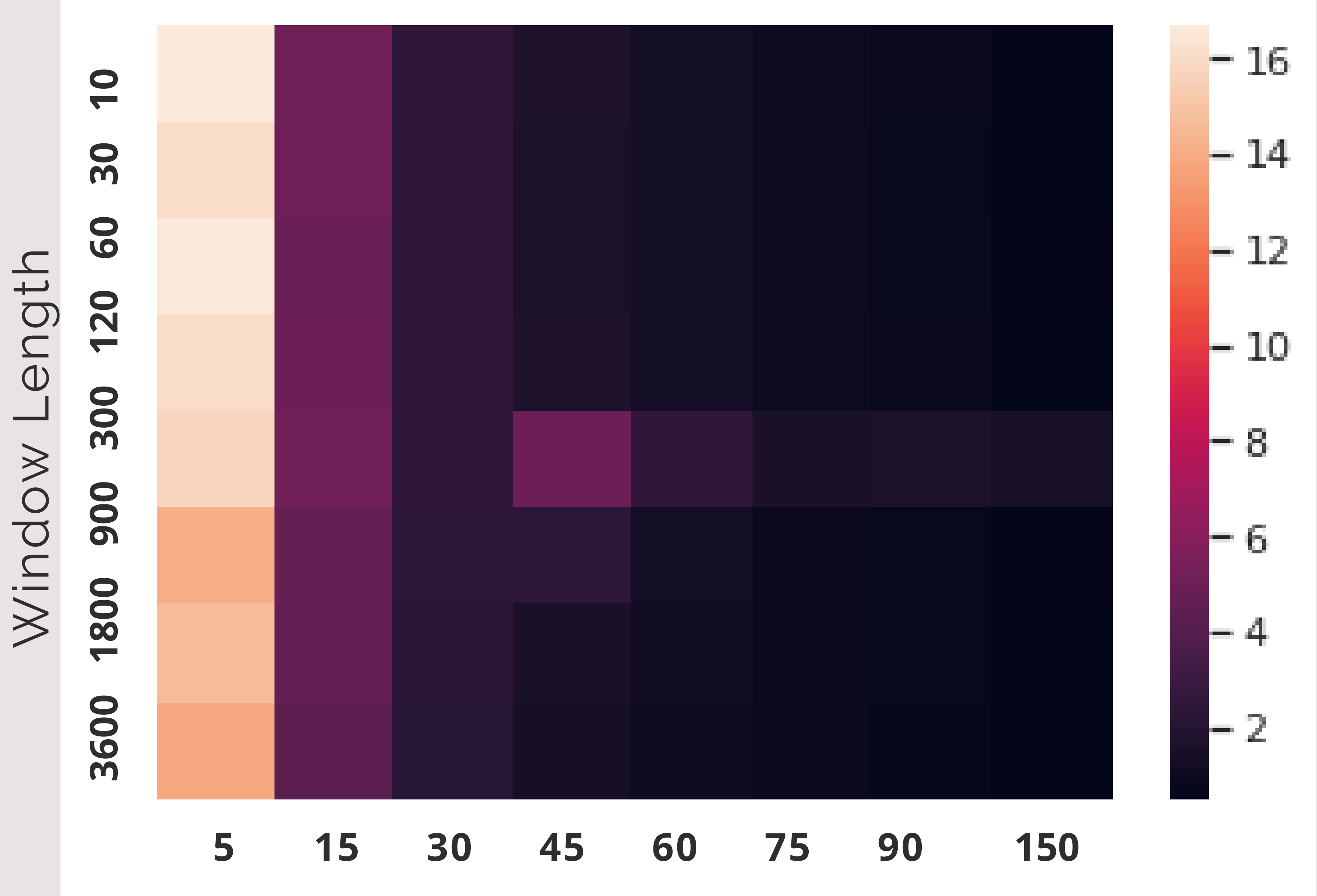



# MLP

## SIZE OF THE MODEL

With the increase of the spacing the size of the model decreases.
With the increase of the windowlength the size of the model decreases.

In both cases: this is due to the dataset having a smaller number of rows



Spacing

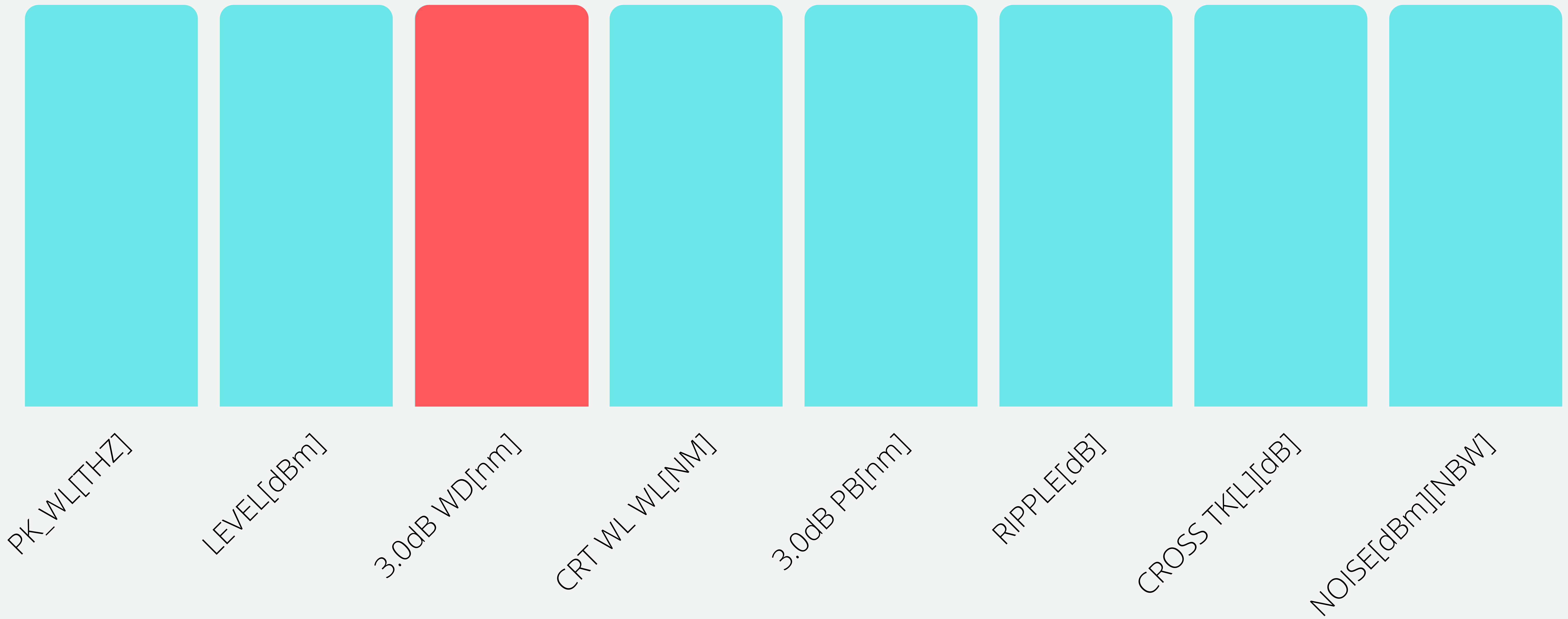
# MLP

## PROCESSING TIME

As expected increasing the Spacing will lead to a smaller dataset. As a consequence the processing time of the MLP decreases

# Feature Removal

Not all the features may be useful



## Select the folder

We select three folders which allow to obtain the best accuracy
(three per classifier, 16 in total)

One of them is randomly selected to compute the feature removal

## The Algorithm

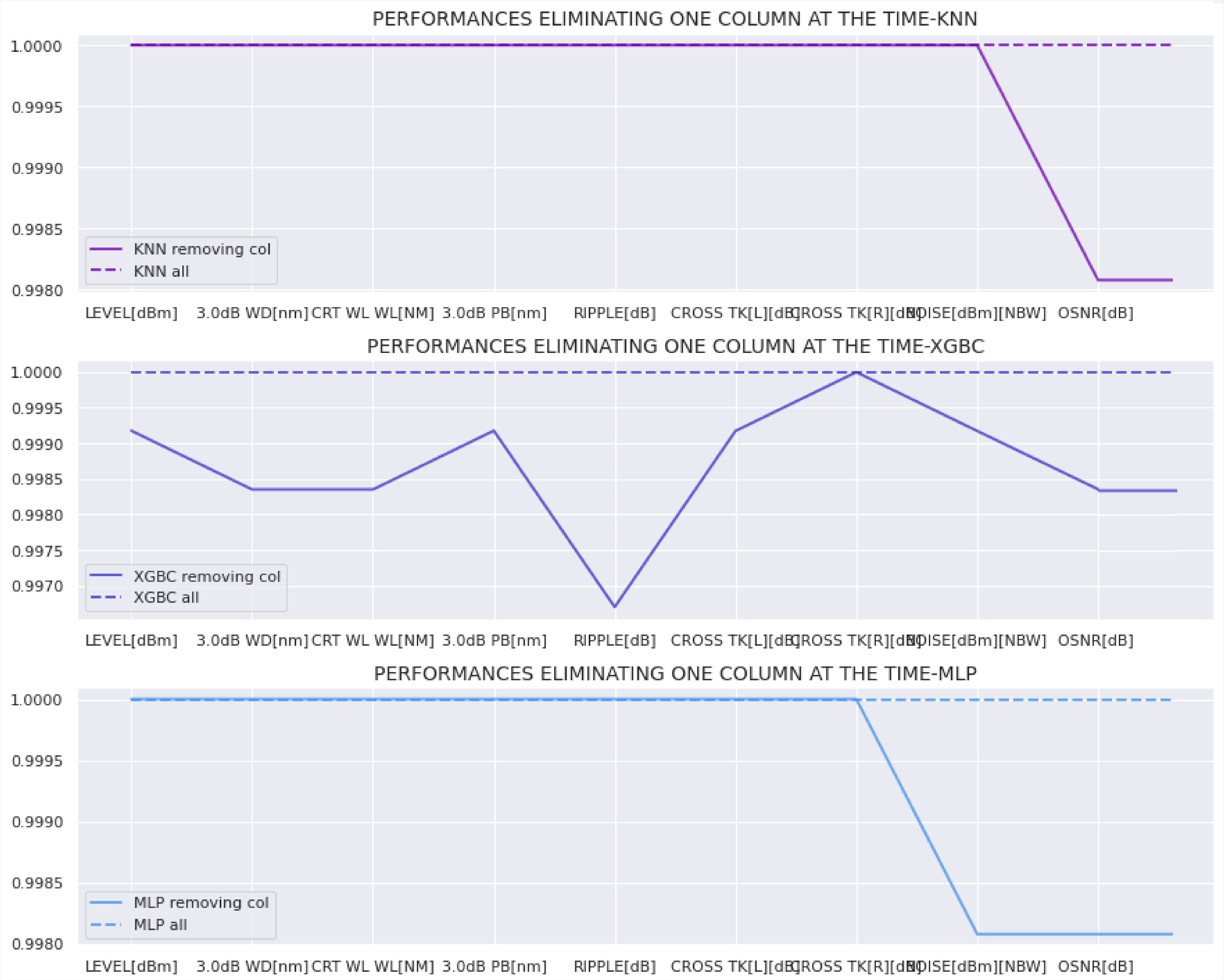
Eliminate one column at the time,
and re-perform the training and testing for each classifier.
Retrieve the accuracy

## Additional considerations

We have tried to eliminate more than one column at the time (2-3-4),
but in these cases the accuracy had the tendency to decrease for all three classifiers.

Concluding, it will be shown only the case in which it has been deleted only ONE column at the time.

Only the useful graphs will be shown below.




# Considerations

It seems that none of the features removed is able to increase the almost perfect accuracy performance of the classifiers

# Accuracy for each Classifier

when only one failure co-exists in the dataset
with the flawless Scenario 0

# Excessive attenuation


## Scenarios

0-1 0-3 0-5
0-2 0-4

The point in the graph represents the mean accuracy.

## Considerations

The accuracy is at its most value, no matter which folder we consider. It's extremely accurate detecting a failure with only scenario_0 present.

# Extra filtering


## Scenarios

0-6 0-7 0-8

The point in the graph represents the mean accuracy.

## Comparison

With respect to the Excessive Attenuation, there is no difference in detecting the faults, even though the attenuation is much higher in the Extra Filtering case.

# EXplainable Artificial Intelligence (XAI)

Optional part - Advanced topics

## WHY DO WE USE XAI?

The need to transform the model black box into a glass-box, understandable by humans, leads us to XAI.

## WHAT IS IT USED FOR?

It helps identifying mistakes made by the model that can be very costly.

**INTERPRETABILITY**

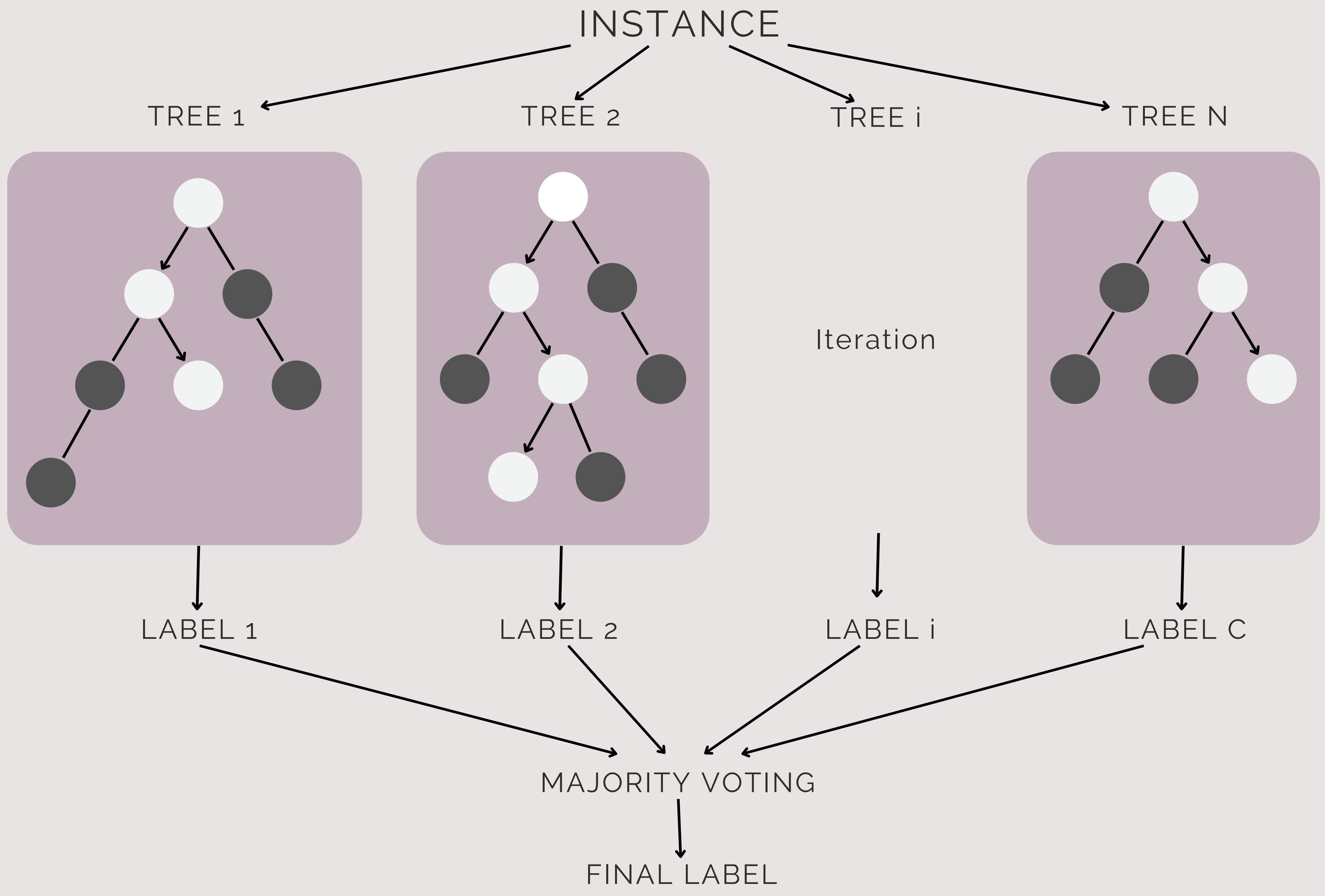
The ability to determine cause and effect from a machine learning model

**EXPLAINABILITY**

The knowledge of both what a node represents and its importance to the model's performance

# Random Forest Classifier

A random forest is a meta estimator that fits a number of decision tree classifiers on various sub-groups of the dataset.
The test dataset is fed to the trees in parallel, the Final Label is addressed through majority voting.



# LIME

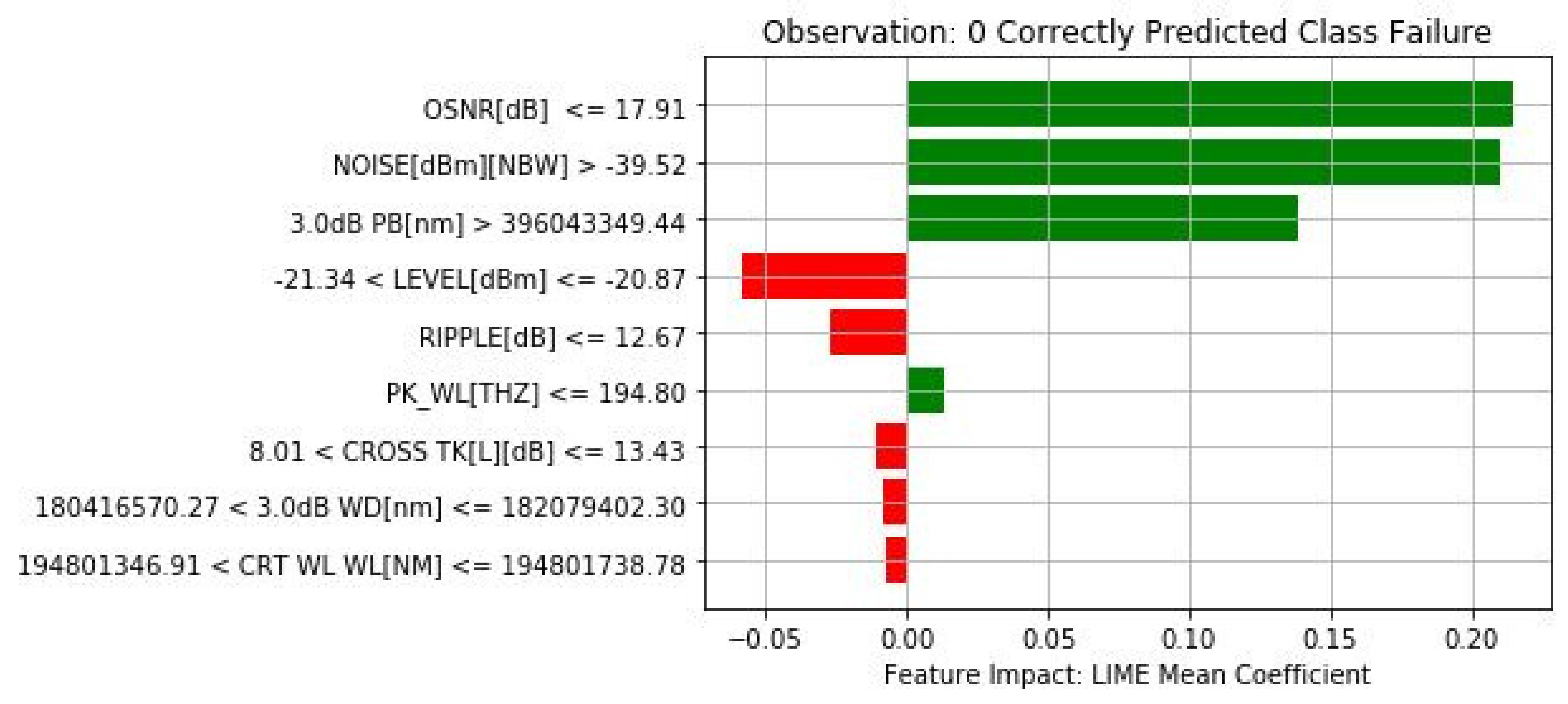
local interpretable
model-agnostic explanation

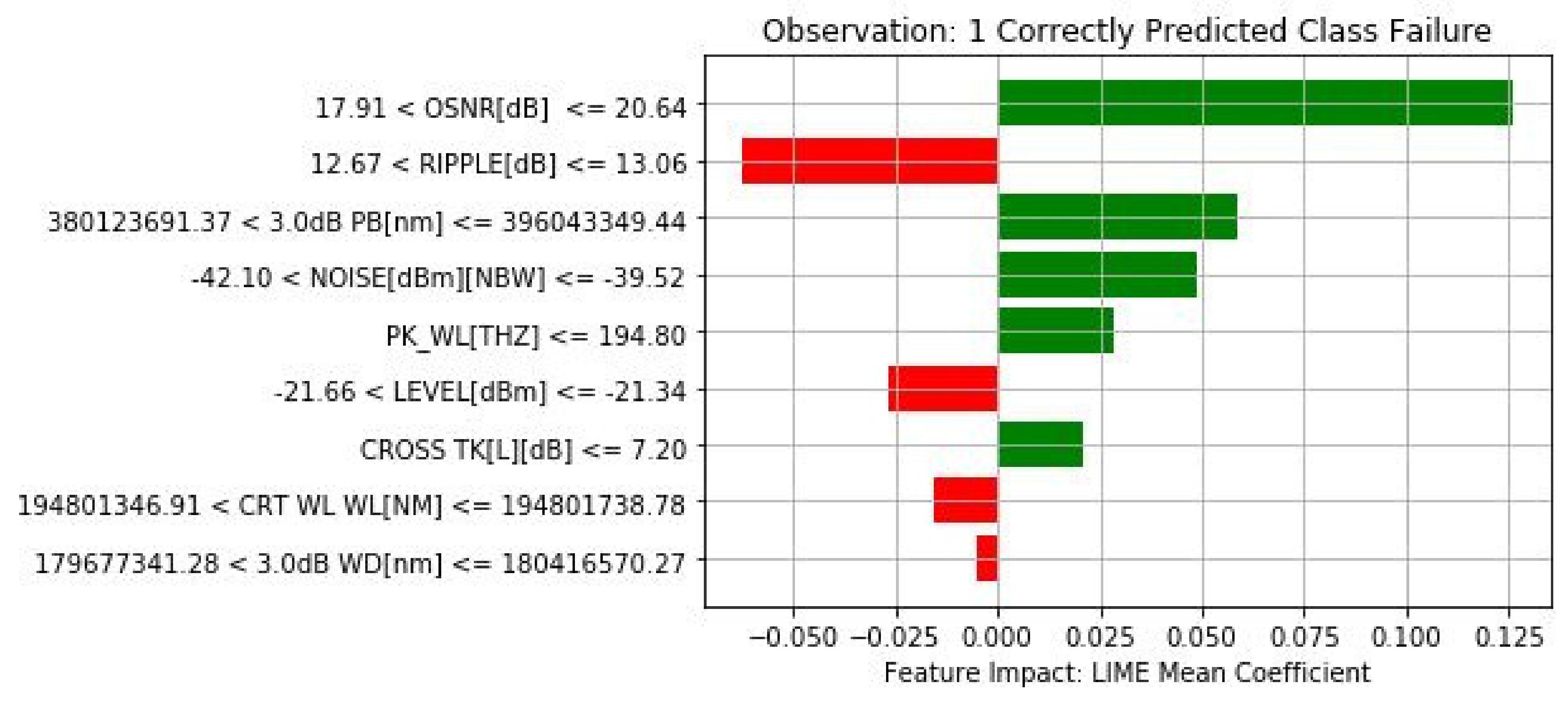
Create a local approximation of the complex model for a specific input (approximation by a linear interpretable model):

$$\xi(x) = \arg\min_{g \in G} L(f, g, \pi_x) + \Omega(g)$$

| | |
|---|---|
| $g \in G$ | Family of linear models |
| $f$ | Blackbox model |
| $g$ | Interpretable model |
| $\pi_x$ | Neighbourhood (proximity measure) |
| $\Omega(g)$ | Measure of simplicity |

The illustrated pictures give a representation of the importance of the features in the dataset, in the prediction of "Failure" scenario. As expected the OSNR[dB] is the most important feature in both observations while coefficients in red have a negative impact in the identification of the faulty scenarios.



# SHAP

## Shapley additive explanations

It's a method born for cooperative game theory, used to estimate the relevance of each feature for individual predictions.

The feature values of a data instance act as players in a coalition.

The Shapley value is the weighted marginal contribution of a feature value across all possible coalitions.

Output with the feature we are interested in

Input data point

Output without the feature we are interested in

#features in coalition

Blackbox model

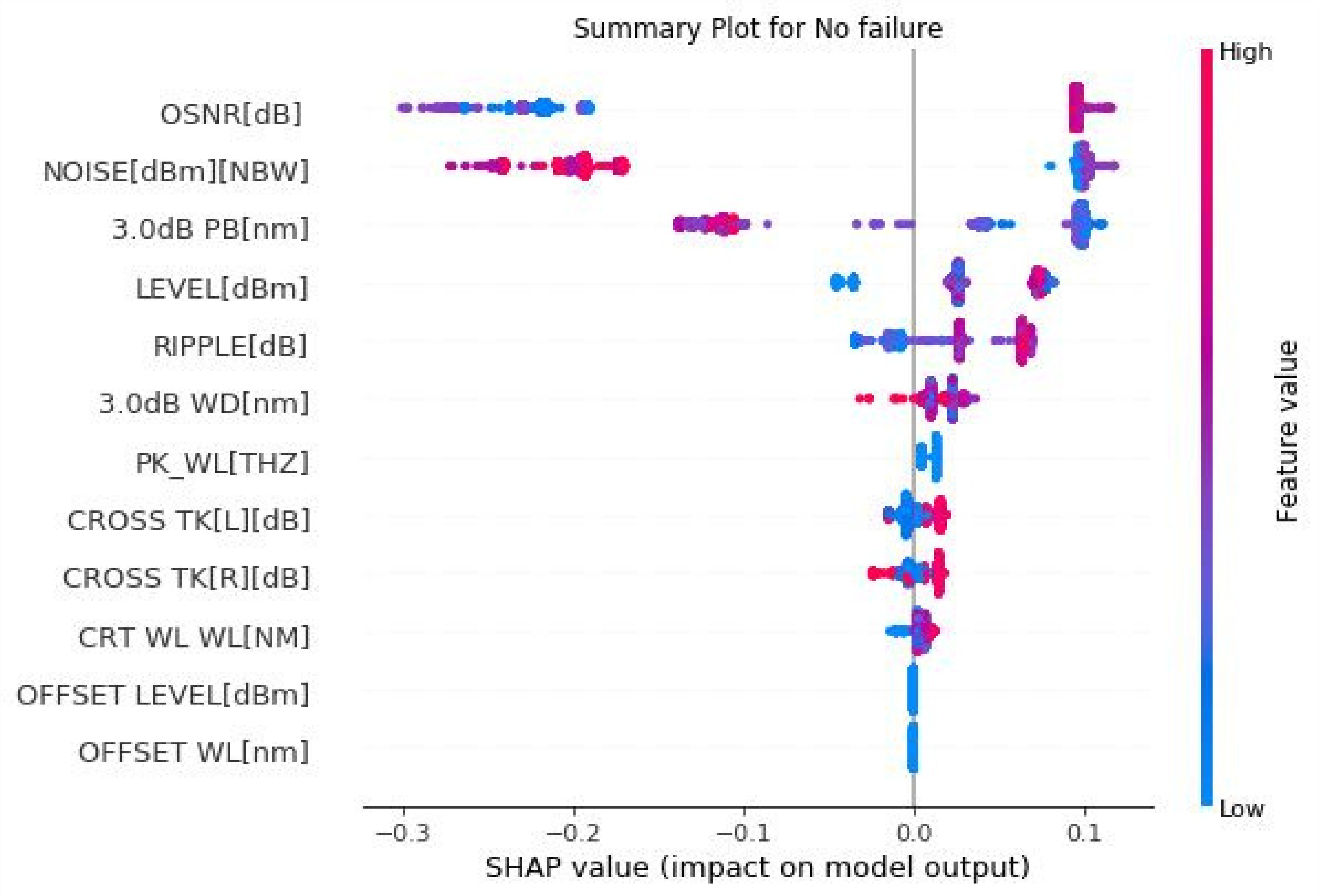
$$\phi_i(f, x) = \sum_{z' \subseteq x} \frac{|z'|!\,(M-|z'|-1)!}{M!} \underbrace{[f_x(z') - f_x(z' \setminus i)]}_{\text{Marginal Value}}$$

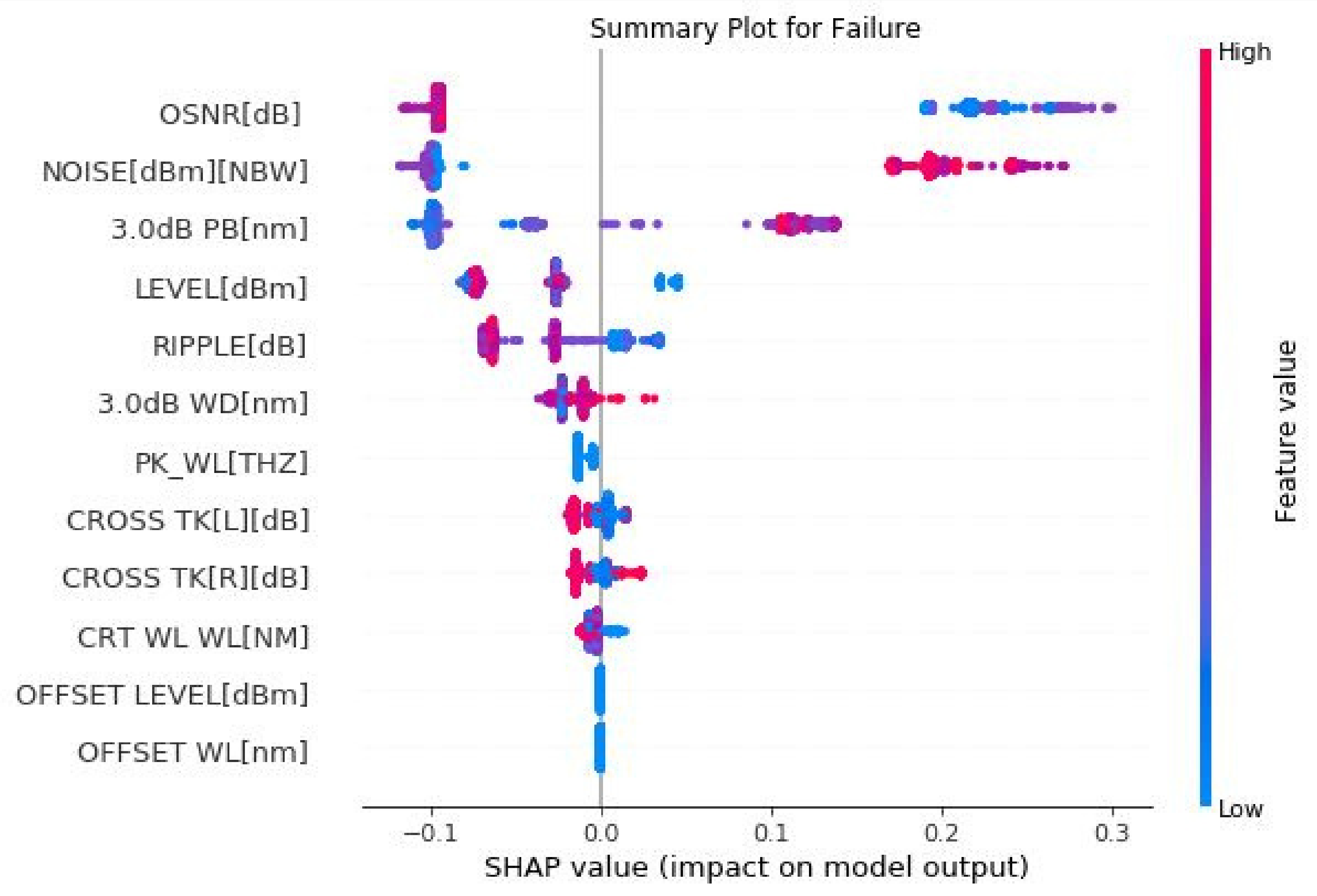
Shapley value for feature i in a given data point

total #features

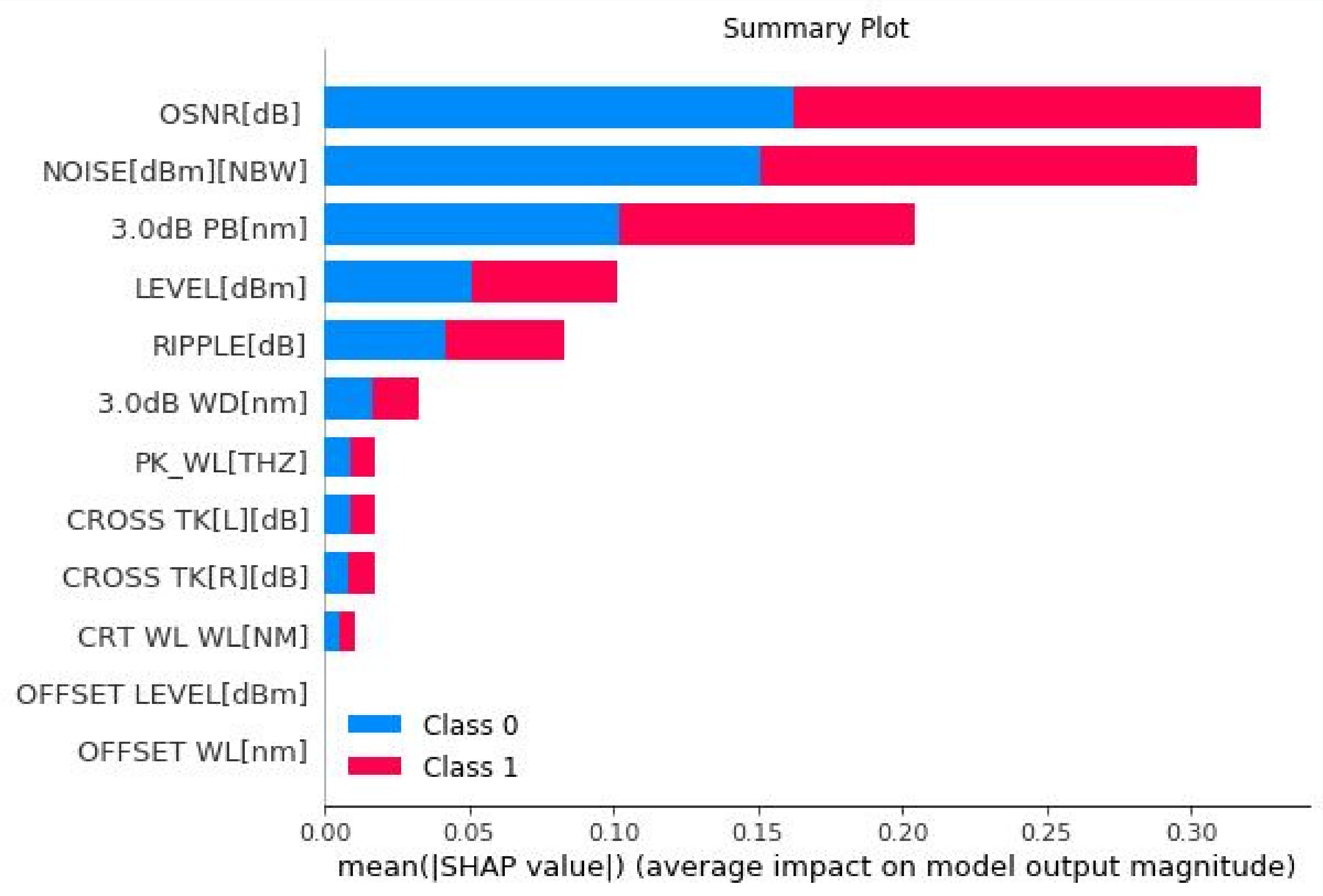
All possible subsets, combinations of features

The interpretation of SHAP plot indicates a high impact of OSNR, NOISE and 3.0dB PB in both failure and no failure recognition.
On the other hand, OFFSET LEVEL and OFFSET WL play minor role in the model decision



The illustrated picture shows the average impact of the coefficients shapley values of the datasets on the model output magnitude. Since the classes are complementary the mean shapley values are the same.



# Thank You For Your Attention.