KU LEUVEN

# Relating Output Symbol Probabilities With Confidence in an End-to-End Speech Recognizer

Narges Baba Ahmadi, Niloufar Baba Ahmadi

Prof. Hugo Van Hamm

KU Leuven - Department of Electrical Engineering (ESAT) – PSI, Belgium

# Introduction

Given the results of a speech recognition network; We evaluated the accuracy of this network's predicted probabilities and found interesting results. The network was uncertain about its predictions although it shouldn't have been as these low probabilities gave us a high accuracy. We tried to solve this problem by giving the prediction patterns to an MLP model and enhancing the probability evaluation per character.

# Data Description

| label | description |
|---|---|
| a. | Spontaneous conversations ('face-to-face') |
| b. | Interviews with teachers of Dutch |
| c. | Spontaneous telephone dialogues (recorded via a switchboard) |
| d. | Spontaneous telephone dialogues (recorded on MD via a local interface) |
| e. | Simulated business negotiations |
| f. | Interviews/discussions/debates (broadcast) |
| g. | (political) Discussions/debates/meetings (non-broadcast) |
| h. | Lessons recorded in the classroom |
| i. | Live (eg sports) commentaries (broadcast) |
| j. | Newsreports/reportages (broadcast) |
| k. | News (broadcast) |
| l. | Commentaries/columns/reviews (broadcast) |
| m. | Ceremonious speeches/sermons |
| n. | Lectures/seminars |
| o. | Read speech |

Table 1.1: Components

The data is a sample of standard dutch spoken in Flanders and the Netherlands, also known as CGN, with approximately a selection of one million words. Different dimensions underlying the variation that can be observed in language use were also taken into account which led to distinguish a number of components.

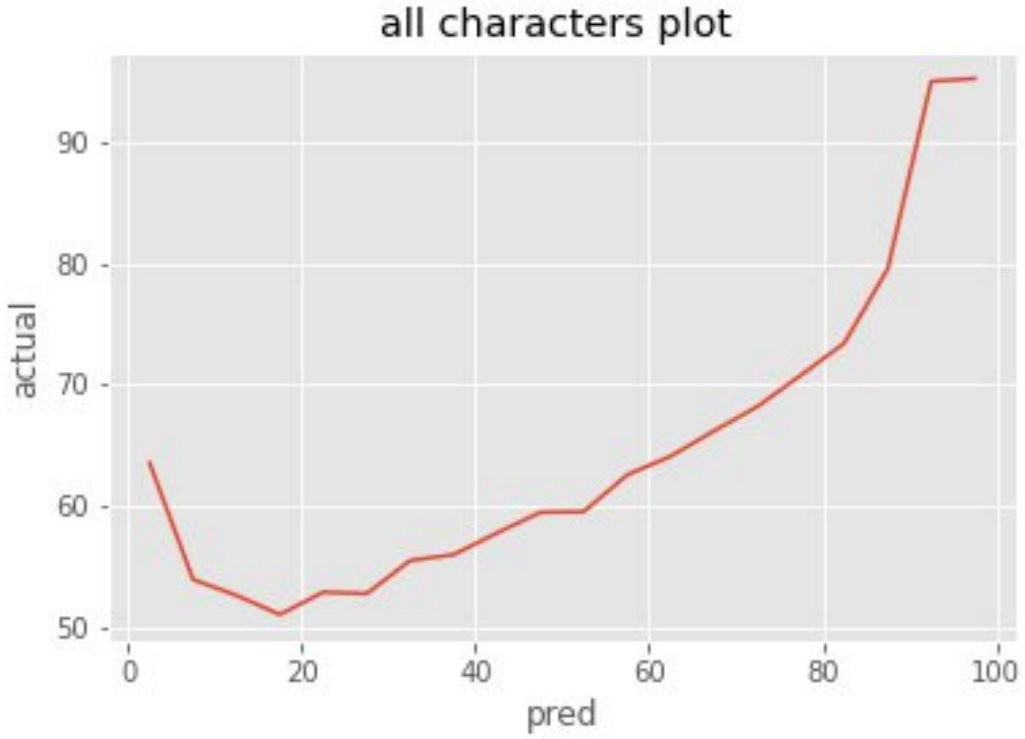
The researchers who trained an ESPnet on CGN data, only used the Flemish data in training and evaluation. Also components c and d were excluded.

# Problem Statement

Each character in a sentence has a certain probability inserted inside an array with 37 elements and each time the network makes a guess, the character with the highest probability is chosen. The result of this network is stored in a 2D numpy array.  The figure bellow illustrates the accuracy comparison of the first-best character in a given interval and the probability given to that character.

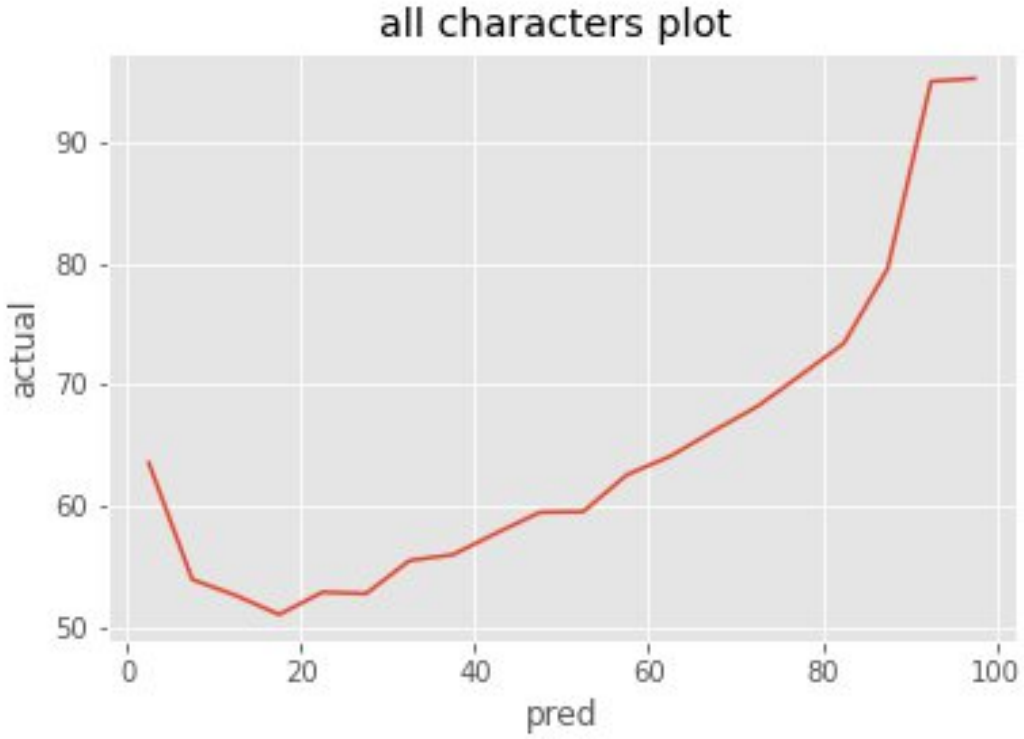
**Objectives:**

- Analyzing the behavior of the posterior probabilities, where does the tail come from? Is the behavior different for larger units ?
- How can we enhance the results of this network? In other words, How to design a network with better confidence scores.
- Which mapping is the best and how to measure that?



# Methods

After going through the the mentioned numpy array, all of the predicted probabilities were found. Next, sclite was used to check whether the predicted character was the correct one. The mismatches found by sclite consist of three different types: Insertion, deletion and substitution. In the scope of this project only deletion and substitution were covered. Using the results of sclite, actual probabilities of characters were calculated.




**There are two problems with this curve:**

- There exists a sharp tail at the beginning of the curve.
- There is a huge gap between the predicted and actual probabilities.

## Where does the tail come from?

- At first we assumed that considering all different components as one could be causing this problem; but Illustrating the same plot for each component showed that all of them had a tail.
- The second assumption was that there might be some characters which are always predicted wrong or right and they might be the reason for the tail, but since almost all character plots had a tail, the hypothesis was once more rejected.
- The last hypothesis says that this tail is caused due to the difference between the amount of data in each interval. Since only a few data points exist in those intervals, the results are not generalized and simply are errors of the network. The figure here illustrates this occurrence.

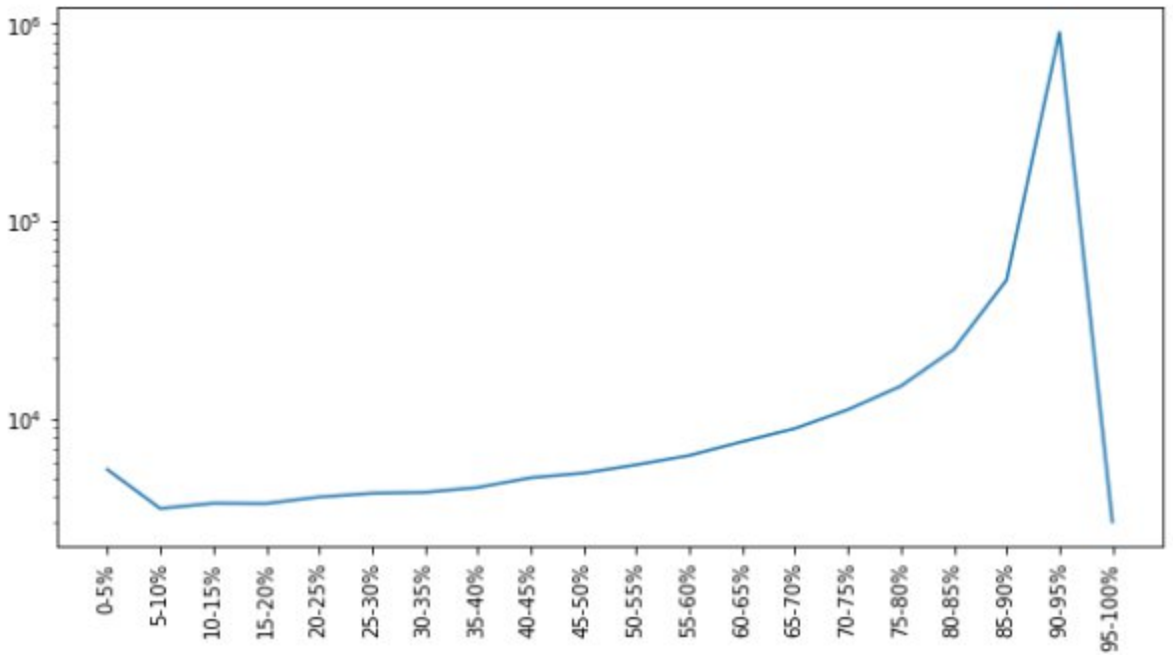
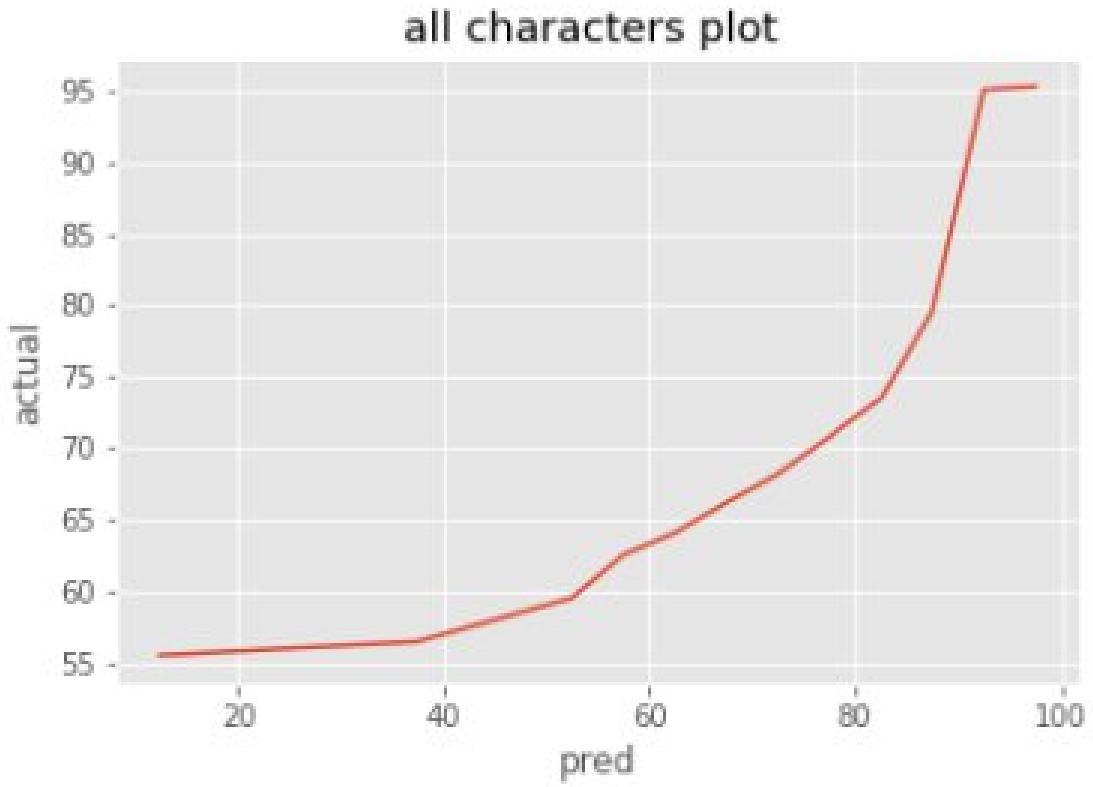



Figure 2.4: Distributions among different intervals

By combining the first intervals, the tail problem got solved in the main all characters figure and in the components and characters plot as well. This solves the problem because the mean of the points were calculated and illustrated on an bigger interval. It is worthwhile mentioning that only 4% of data exists before the 55% interval. Any data point under 55% is an error of the network.



# Dataset Construction

To enhance the results of the original network, we decided to train an MLP network but first the network's features and targets needed to be constructed.
The features were extracted from the predicted probabilities; to put it in another way, the outputs of the previous network are the inputs of the new one.
To create the targets of the network, the whole CGN data set was examined and look-up table 2.3 was formed.
For each index of each record, first we found its rank among the 37 elements of the record and after finding the corresponding interval of that number and using table 2.3, we replaced the value of that index with the correct value.

| | 0-5% | 5-10% | 10-15% | 15-20% | 20-25% | 25-30% | 30-35% | 35-40% | 40-45% | 45-50% |
|---|---|---|---|---|---|---|---|---|---|---|
| First best | 0.63562 | 0.53960 | 0.52623 | 0.51052 | 0.52902 | 0.52753 | 0.55518 | 0.55991 | 0.57848 | 0.59485 |
| Second best | 0.00035 | 0.00169 | 0.00267 | 0.00240 | 0.00175 | 0.00272 | 0.00304 | 0.00209 | 0.00376 | ['N'] |
| Third best | 0.00035 | 0.00210 | 0.00237 | 0.00263 | 0.00314 | ['N'] | ['N'] | ['N'] | ['N'] | ['N'] |
| 4th best | 0.00032 | 0.00153 | 0.00240 | ['N'] | ['N'] | ['N'] | ['N'] | ['N'] | ['N'] | ['N'] |
| 5th best | 0.00028 | 0.00089 | 0.00719 | ['N'] | ['N'] | ['N'] | ['N'] | ['N'] | ['N'] | ['N'] |
| 6th best | 0.00022 | ['N'] | ['N'] | ['N'] | ['N'] | ['N'] | ['N'] | ['N'] | ['N'] | ['N'] |
| 7th best | 0.00019 | ['N'] | ['N'] | ['N'] | ['N'] | ['N'] | ['N'] | ['N'] | ['N'] | ['N'] |
| 8th best | 0.00018 | ['N'] | ['N'] | ['N'] | ['N'] | ['N'] | ['N'] | ['N'] | ['N'] | ['N'] |
| 9th best | 0.00015 | ['N'] | ['N'] | ['N'] | ['N'] | ['N'] | ['N'] | ['N'] | ['N'] | ['N'] |
| 10th line | 0.00012 | ['N'] | ['N'] | ['N'] | ['N'] | ['N'] | ['N'] | ['N'] | ['N'] | ['N'] |
| 11th best | 0.0.00010 | ['N'] | ['N'] | ['N'] | ['N'] | ['N'] | ['N'] | ['N'] | ['N'] | ['N'] |
| 12th best | 0.00010 | ['N'] | ['N'] | ['N'] | ['N'] | ['N'] | ['N'] | ['N'] | ['N'] | ['N'] |
| 13th best | 8.7861e-05 | ['N'] | ['N'] | ['N'] | ['N'] | ['N'] | ['N'] | ['N'] | ['N'] | ['N'] |
| 14th best | 7.1971e-05 | ['N'] | ['N'] | ['N'] | ['N'] | ['N'] | ['N'] | ['N'] | ['N'] | ['N'] |
| 15th best | 7.7579e-05 | ['N'] | ['N'] | ['N'] | ['N'] | ['N'] | ['N'] | ['N'] | ['N'] | ['N'] |
| 16th best | 5.7016e-05 | ['N'] | ['N'] | ['N'] | ['N'] | ['N'] | ['N'] | ['N'] | ['N'] | ['N'] |
| 17th best | 6.0754e-05 | ['N'] | ['N'] | ['N'] | ['N'] | ['N'] | ['N'] | ['N'] | ['N'] | ['N'] |
| 18th best | 5.3277e-05 | ['N'] | ['N'] | ['N'] | ['N'] | ['N'] | ['N'] | ['N'] | ['N'] | ['N'] |
| 19th best | 4.3930e-05 | ['N'] | ['N'] | ['N'] | ['N'] | ['N'] | ['N'] | ['N'] | ['N'] | ['N'] |
| 20th best | 6.0754e-05 | ['N'] | ['N'] | ['N'] | ['N'] | ['N'] | ['N'] | ['N'] | ['N'] | ['N'] |
| 21th best | 5.7950e-05 | ['N'] | ['N'] | ['N'] | ['N'] | ['N'] | ['N'] | ['N'] | ['N'] | ['N'] |
| 22th best | 6.1689e-05 | ['N'] | ['N'] | ['N'] | ['N'] | ['N'] | ['N'] | ['N'] | ['N'] | ['N'] |
| 23th best | 5.7016e-05 | ['N'] | ['N'] | ['N'] | ['N'] | ['N'] | ['N'] | ['N'] | ['N'] | ['N'] |
| 24th best | 5.8885e-05 | ['N'] | ['N'] | ['N'] | ['N'] | ['N'] | ['N'] | ['N'] | ['N'] | ['N'] |
| 25th best | 6.9167e-05 | ['N'] | ['N'] | ['N'] | ['N'] | ['N'] | ['N'] | ['N'] | ['N'] | ['N'] |
| 26th best | 6.6363e-05 | ['N'] | ['N'] | ['N'] | ['N'] | ['N'] | ['N'] | ['N'] | ['N'] | ['N'] |
| 27th best | 7.7579e-05 | ['N'] | ['N'] | ['N'] | ['N'] | ['N'] | ['N'] | ['N'] | ['N'] | ['N'] |
| 28th best | 6.4493e-05 | ['N'] | ['N'] | ['N'] | ['N'] | ['N'] | ['N'] | ['N'] | ['N'] | ['N'] |
| 29th best | 6.9167e-05 | ['N'] | ['N'] | ['N'] | ['N'] | ['N'] | ['N'] | ['N'] | ['N'] | ['N'] |
| 30th best | 7.2905e-05 | ['N'] | ['N'] | ['N'] | ['N'] | ['N'] | ['N'] | ['N'] | ['N'] | ['N'] |
| 31th best | 7.2905e-05 | ['N'] | ['N'] | ['N'] | ['N'] | ['N'] | ['N'] | ['N'] | ['N'] | ['N'] |
| 32th best | 6.5428e-05 | ['N'] | ['N'] | ['N'] | ['N'] | ['N'] | ['N'] | ['N'] | ['N'] | ['N'] |
| 33th best | 7.2905e-05 | ['N'] | ['N'] | ['N'] | ['N'] | ['N'] | ['N'] | ['N'] | ['N'] | ['N'] |
| 34th best | 0.0001 | ['N'] | ['N'] | ['N'] | ['N'] | ['N'] | ['N'] | ['N'] | ['N'] | ['N'] |
| 35th best | 9.1599e-05 | ['N'] | ['N'] | ['N'] | ['N'] | ['N'] | ['N'] | ['N'] | ['N'] | ['N'] |
| 36th best | 9.9077e-05 | ['N'] | ['N'] | ['N'] | ['N'] | ['N'] | ['N'] | ['N'] | ['N'] | ['N'] |
| 37th best | 0.0001 | ['N'] | ['N'] | ['N'] | ['N'] | ['N'] | ['N'] | ['N'] | ['N'] | ['N'] |

| | 50-55% | 55-60% | 60-65% | 65-70% | 70-75% | 75-80% | 80-85% | 85-90% | 90-95% | 95-100% |
|---|---|---|---|---|---|---|---|---|---|---|
| First best | 0.59556 | 0.62573 | 0.64099 | 0.66208 | 0.68199 | 0.70791 | 0.73446 | 0.79555 | 0.95009 | 0.95252 |
| Second best | ['N'] | ['N'] | ['N'] | ['N'] | ['N'] | ['N'] | ['N'] | ['N'] | ['N'] | ['N'] |

* The header row represents the predicted probabilities.
* The cells represent the actual probability.
* In the context of this table, ['N'] means null.
* The rest of the rows are all null after 50%.

Table 2.3: Look-up table

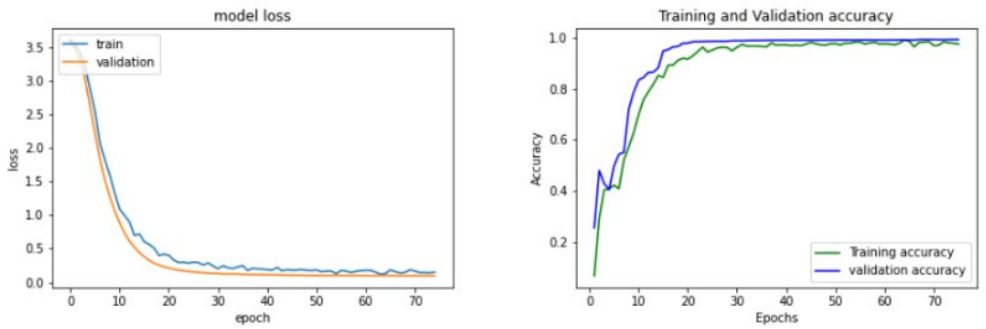
# Models

The model we trained was a sequential model made of four layers. Figure 2.8 illustrates the model. The first three layers had Relu activation function and the last layer had Softmax as the activation function. Categorical cross-entropy was used as the loss function and the optimizer was set to Adam. The models were trained on normalized data.

```
Model: "sequential_16"
_________________________________________________________________
 Layer (type)                 Output Shape              Param #
=================================================================
 dense_107 (Dense)            (None, 120)               4560

 dropout_42 (Dropout)         (None, 120)               0

 dense_108 (Dense)            (None, 60)                7260

 dropout_43 (Dropout)         (None, 60)                0

 dense_109 (Dense)            (None, 30)                1830

 dropout_44 (Dropout)         (None, 30)                0

 dense_110 (Dense)            (None, 37)                1147

=================================================================
Total params: 14,797
Trainable params: 14,797
Non-trainable params: 0
_________________________________________________________________
```

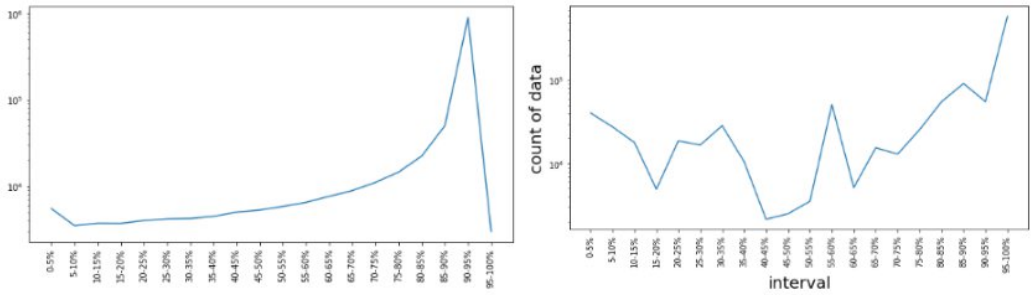
Figure 2.8: Model description

## Cross-entropy loss




Model's Loss Model's accuracy

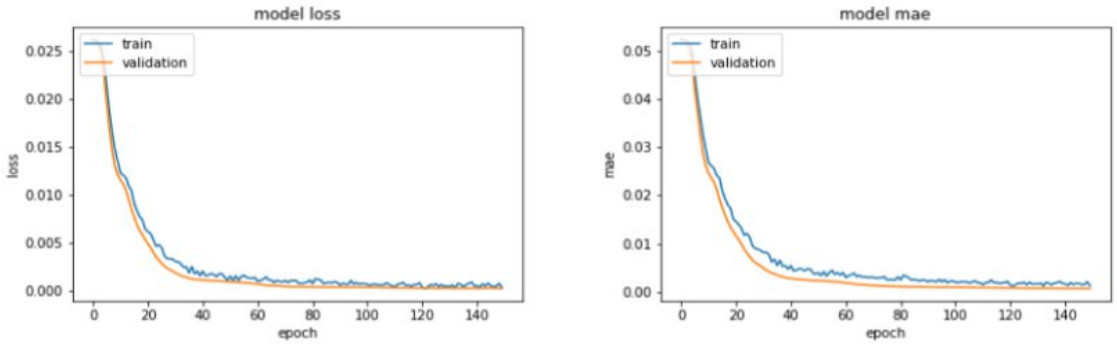
Figure 2.11: Loss and accuracy of the model with cross-entropy loss function trained on nomalized data




(i) Distribution before training (ii) Distribution after training

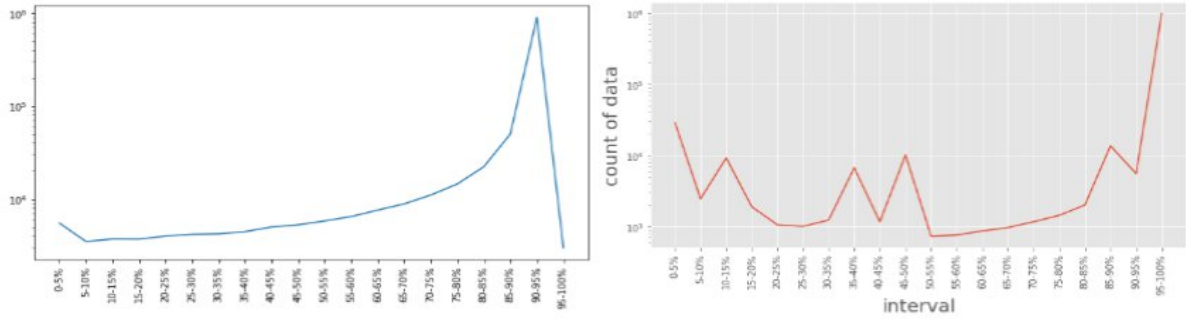
Figure 2.12: Distribution comparison in the model with Cross-entropy loss trained on normalized data

## Mean Square Error loss




Model's MSE Model's MAE

Figure 2.16: MSE and MAE of the regression model on normalized data




(i) Distribution before training (ii) Distribution after training

Figure 2.17: Distribution comparison in the regression model (Normalised data)

# Kullback–Leibler divergence

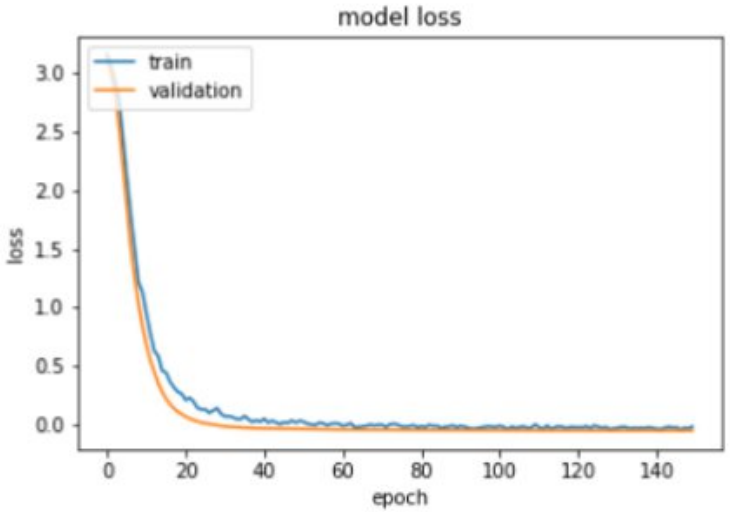
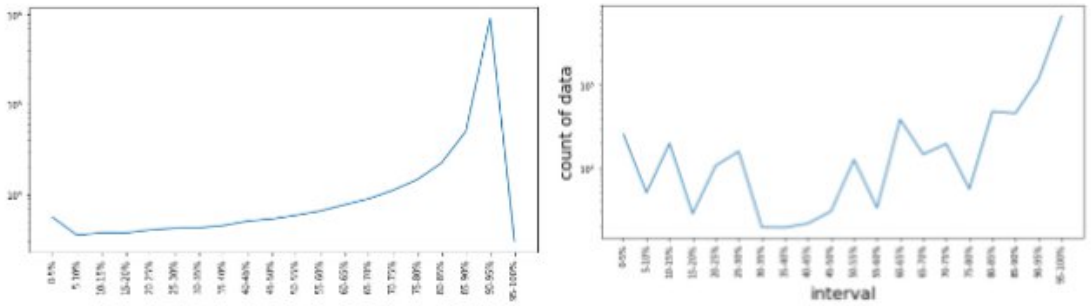



Figure 2.19: Loss of the model with KLD loss function




(a) Distribution before training

(b) Distribution after training

Figure 2.20: Distribution comparison in the model with KLD loss function

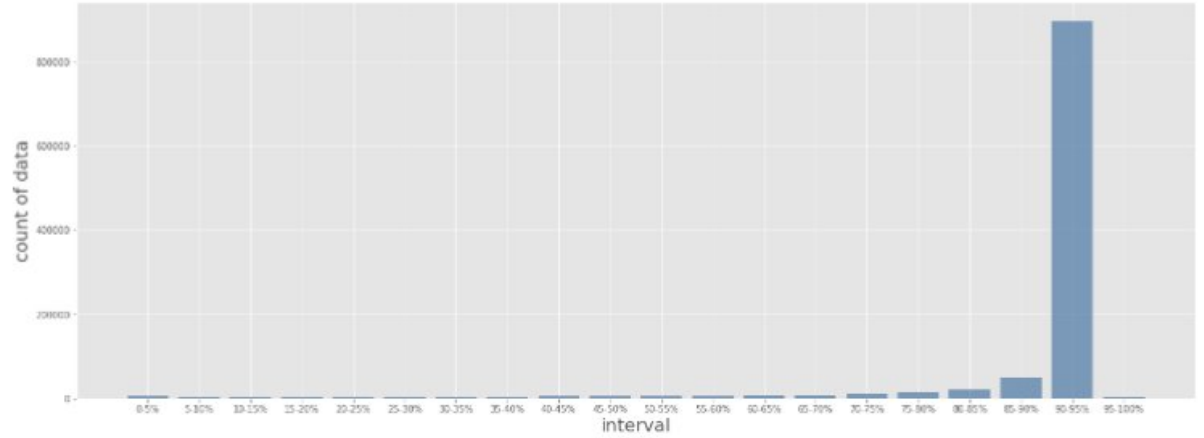
# Data Augmentation

As you can see in the models, the upper intervals are getting better results and lower ones are getting worse.

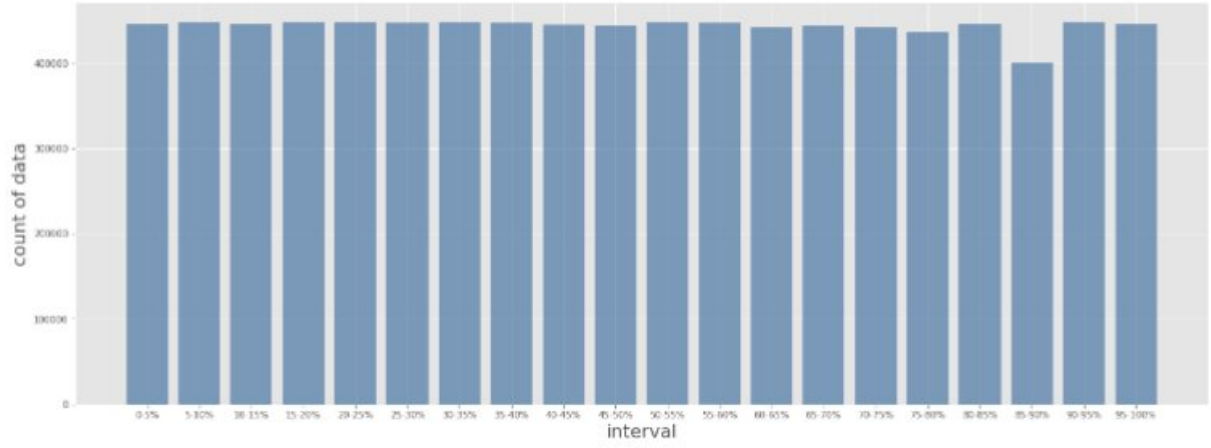
We assumed that low distribution of lower intervals is causing this problem so we decided to add some data to the lower intervals.

We increased the number of data in lower intervals by duplicating the previous data points in each interval that had a low distribution.

After changing the data set, we trained the same models on the new data set.



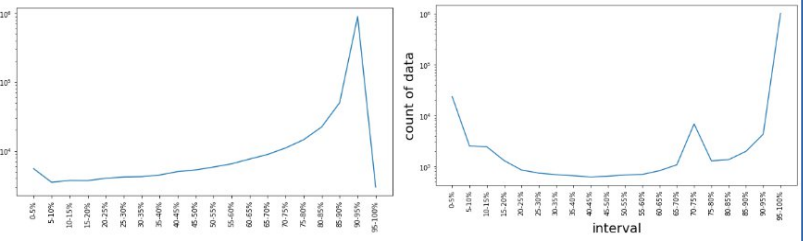

(a) Distribution before data augmentation




(b) Distribution after data augmentation

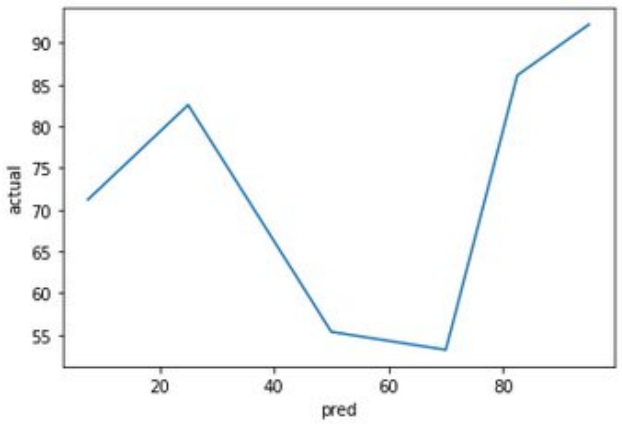
Figure 2.21: Distribution comparison before and after data augmentation

## Cross-entropy loss

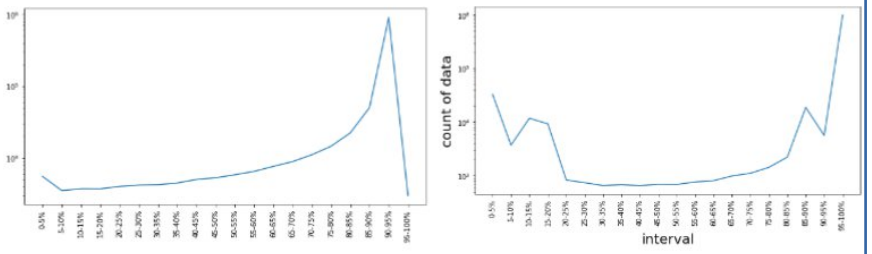



(a) Distribution before training (b) Distribution after training

Figure 2.23: Distribution comparison in the model with KLD loss function

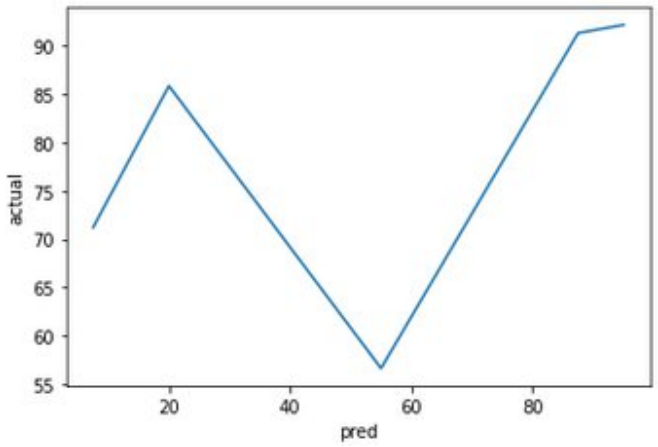



## Mean Square Error loss

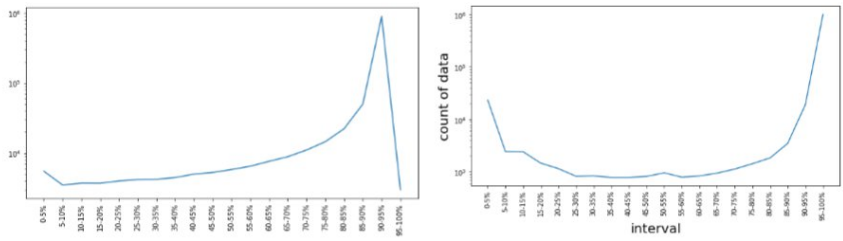



(a) Distribution before training (b) Distribution after training

Figure 2.23: Distribution comparison in the model with MSE loss function (Data Augmentation section)

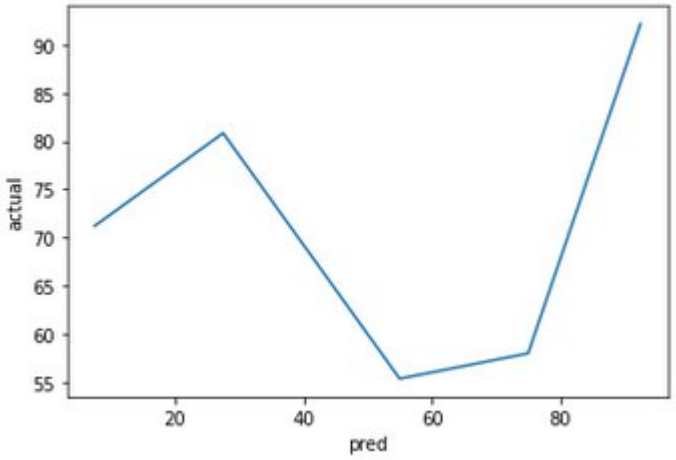



## Kullback–Leibler divergence




(a) Distribution before training (b) Distribution after training

Figure 2.25: Distribution comparison in the model with KLD loss function (Data Augmentation section)

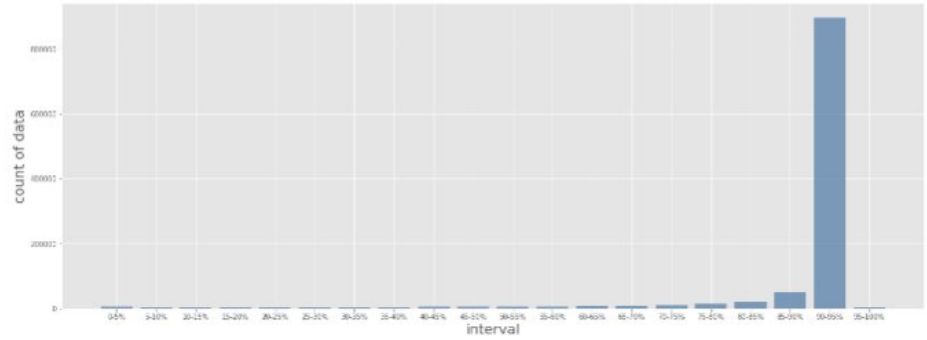




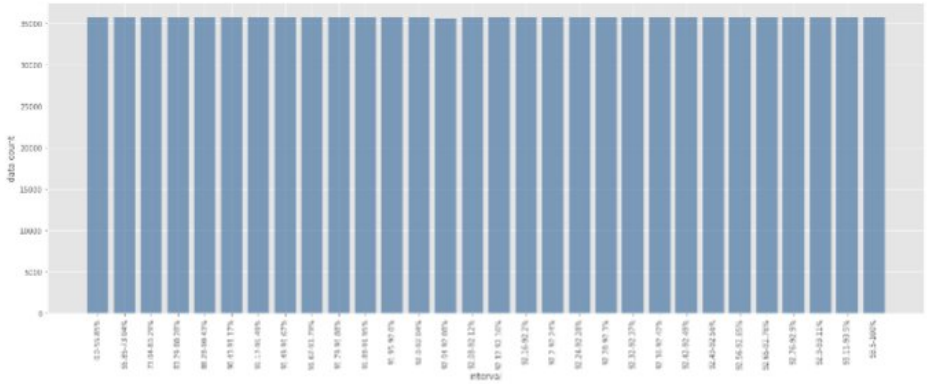

KU LEUVEN

# Dataset Construction (Second approach)

Instead of copying the data to have intervals with equal lengths, we tried to divide intervals based on data distribution.



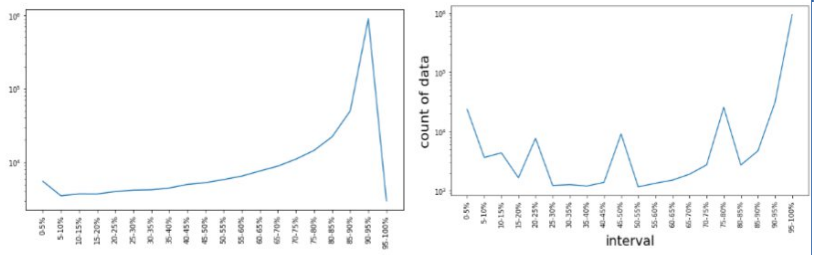

(a) Distribution before reconstruction




(b) Distribution after reconstruction

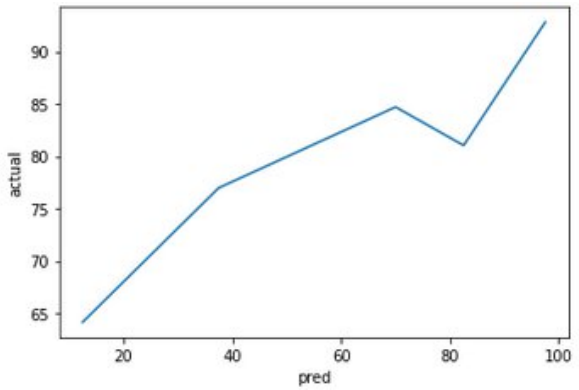
Figure 2.26: Distribution comparison before and after reconstruction

## Cross-entropy loss

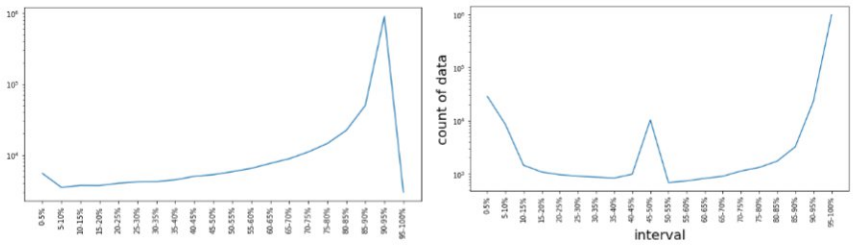



(a) Distribution before training (b) Distribution after training

Figure 2.28: Distribution comparison in the model with Cross-entropy loss functio (Data Construction 2 section)

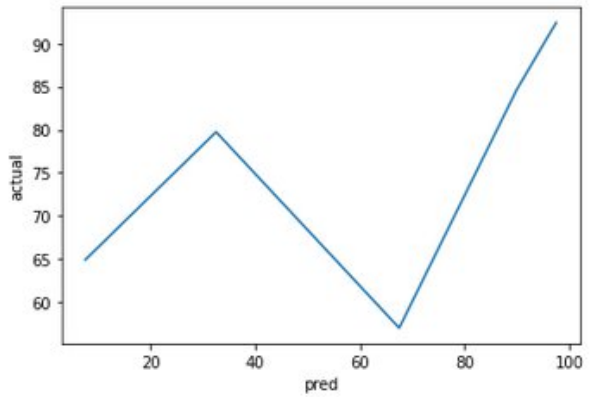



## Mean Square Error loss

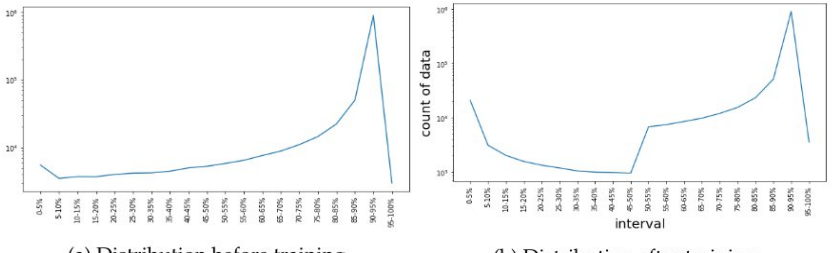



(a) Distribution before training (b) Distribution after training

Figure 2.30: Distribution comparison in the model with MSE loss function (Data Construction 2 section)

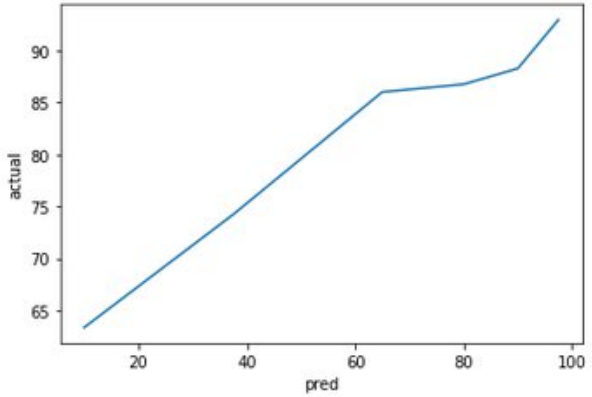



## Kullback–Leibler divergence




(a) Distribution before training (b) Distribution after training

Figure 2.40: Distribution comparison in the model with KLD loss function and unnormalized data (Data Construction 2 section)

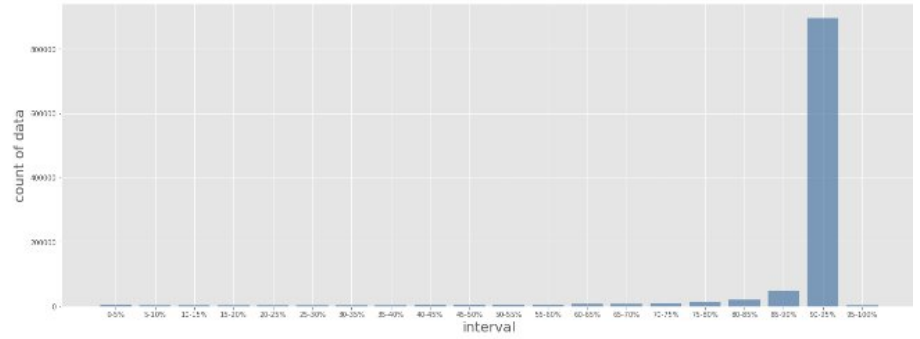




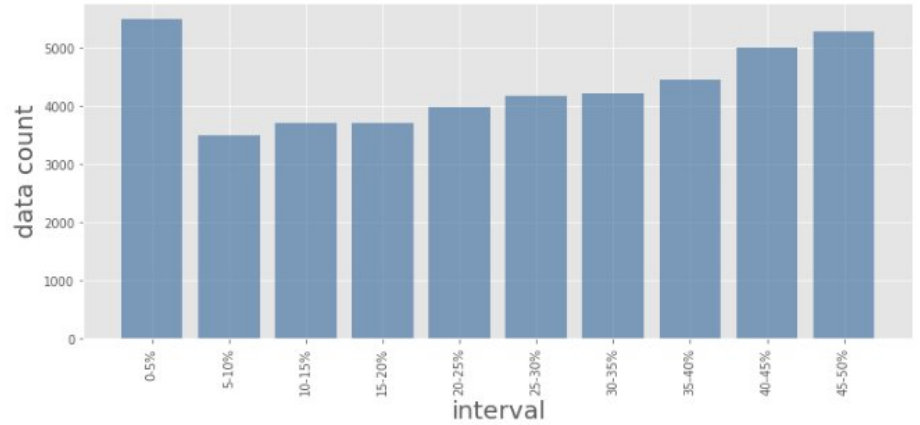

KU LEUVEN

# Dataset Construction (Third approach)

Since the most of the problem stem from the first half of the data set meaning under 50%, we split the whole data set with the threshold of 50% and trained the network on first half.




(a) Distribution before reconstruction

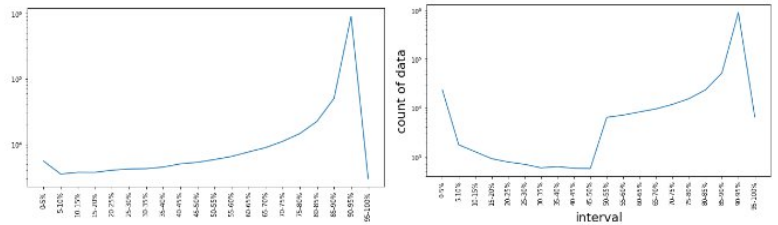



(b) Distribution after reconstruction

Figure 2.41: Distribution comparison before and after reconstruction

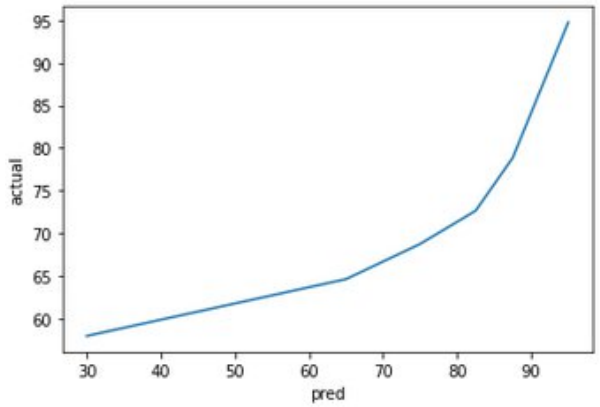
## Cross-entropy loss

As evident from figure 2.43, 3.53% of data exist under 55% threshold.



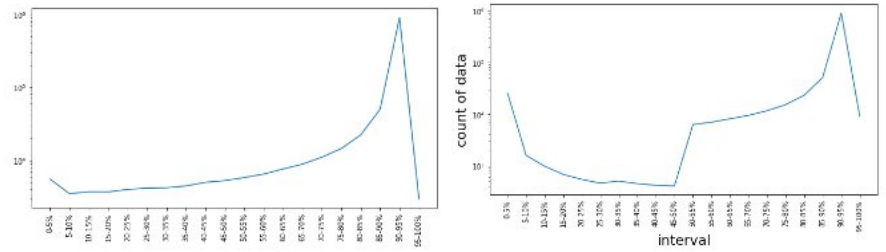

(a) Distribution before training (b) Distribution after training

Figure 2.43: Distribution comparison in the model with Cross-entropy loss function (Data Construction 3 section)



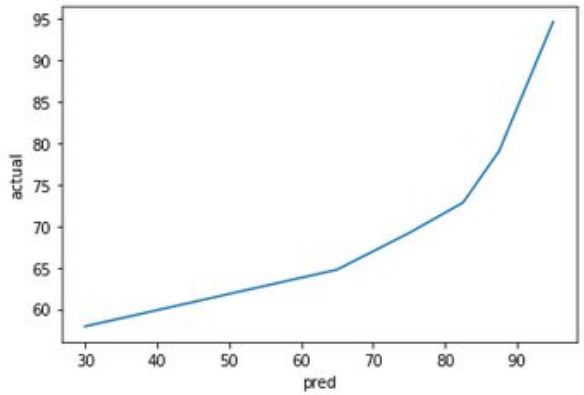

## Mean Square Error loss

As evident from figure 2.47, 3.50% of data exist under 55% threshold.



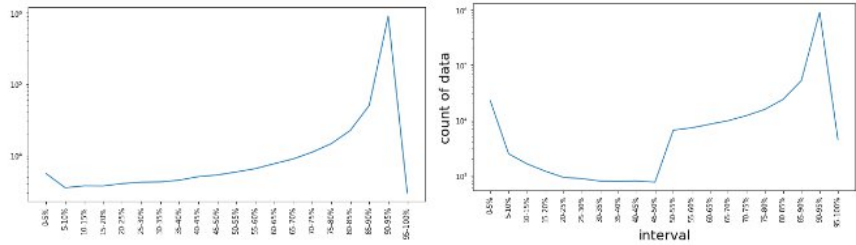

(a) Distribution before training (b) Distribution after training

Figure 2.47: Distribution comparison in the model with MSE loss function (Data Construction 3 section)



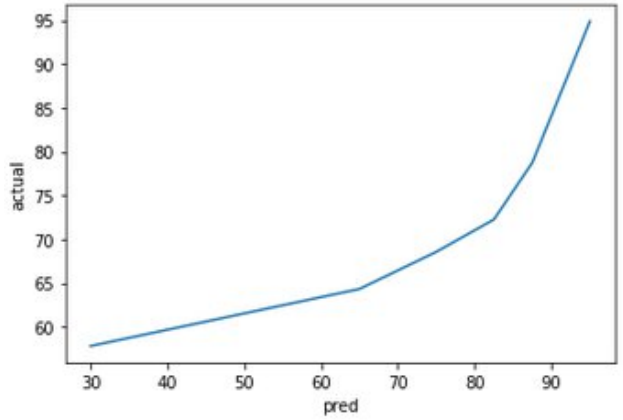

## Kullback–Leibler divergence

As evident from figure 2.51, 3.63% of data exist under 55% threshold.




(a) Distribution before training (b) Distribution after training

Figure 2.51: Distribution comparison in the model with KLD loss function (Data Construction 3 section)



# Data Augmentation

# Q & A

Thank you.