First, we look at the diabetes dataset and preprocess it before continuing to the training process. We decided to use min-max normalization and we normalize the entire dataset with SKLearn function MinMaxScaler.  
The use of the MinMaxScaler with PCA resulted in a more easy-to-read plot where each group of sick/not sick points are divided clearly.  
Chart, scatter chart

Description automatically generated

Second, we train both generator and discriminator with the following architecture for each of the models:

Generator:

Output  
[32, 9]

Dense  
[32, 40]

Dense  
[32, 20]

Dense  
[32, 10]

Input  
[32, 5]

Discriminator:

Output  
[32, 1]

Dense  
[32, 10]

Dense  
[32, 20]

Dense  
[32, 40]

Input  
[32, 9]

Dropout  
0.1

Dropout  
0.1

After the training process we give the generator random noise and receive an input data that is similar to the normalized data that we train the model on, that is not the intended output and thus we use the reverse\_transform method and receive a new dataset within the correct range of values.  
Following that, we let the discriminator guess if these samples are real or not and we label the discriminator guess for real samples as 1 and 0 otherwise. Table

Description automatically generated

Right now we can definitely see that the data generated in a similar “pattern” and share the same characteristics as the original data, age range is logical and even pregnancies **but** we need to check if the diabetes label is similar, meaning that people who have diabetes share the same characteristics between the generated and original data.

**Original data**

**Text

Description automatically generated**

**Generated data**

**Text, table

Description automatically generated with medium confidence**

We see similar behavior for plas and insu but that is not enough, we can plot both data points at a 2D graph with the use of PCA and see how both groups distribute.  
On the left, positive samples and on the right, negative samples, both data points are **not** normalized and are the raw values presented in the tables above.

Chart, scatter chart

Description automatically generatedChart, scatter chart

Description automatically generated

The graph below shows the normalized data and the normalized generated data and we can see Chart, scatter chart

Description automatically generated

Overall, the samples are close to the original but we do see an odd behavior at the left side of the plots with the line of dots, unfortunately we are not sure why this behavior is existed but that is the original data plotted with PCA so the dimension reduction might be the cause.

As of our discriminator, we want to check how many samples out of the 100 samples did fool the discriminator. We find out the 58 samples has been classified as true out of the 100 generated samples!

**Euclidean distance:**

We calculated the distance between positive examples on the real data and between the generated data to the real data and found out the following [Positive samples, Negative samples]:

Distance between real data points – Mean [160.30, 114.76]  
Distance between generated to real data points – Mean [185.30, 88.83]

The results are very similar in terms of distance between points for the real data compared to the generated data.

**Loss graph:**

We do see a “back and forth” motion in the losses graph but it is not clear whether they switched places during this period, most of the times the clear “winner” was the discriminator and we think it is related to the ease of classification task in that particular dataset.

Chart

Description automatically generated