Predicting Survival of Breast Cancer Patients after Operation

Neural Networks Lab Project

Prof. Konstantinos Karampidis

By Anna Manucharyan

27 May, 2023

Introduction

The goal of this project is to develop a predictive model using neural networks to determine the survival of breast cancer patients after undergoing an operation. The dataset used for this project is the "Haberman's Survival" dataset, which is from the UCI Machine Learning Repository. It consists of data collected from patients who underwent surgery for breast cancer at the University of Chicago's Billings Hospital between 1958 and 1970.

The dataset contains four primary attributes:

Age: The age of the patient at the time of the operation.

Year: The year in which the operation took place.

Nodes: The number of positive axillary nodes detected.

Status: The survival status of the patient after the operation (1 = survived for 5 years or longer, 2 = died within 5 years).

I tried to build a predictive model that can determine the survival of breast cancer patients based on their age, year of operation, and the number of positive axillary nodes detected. By analyzing these attributes, we aim to develop a model that can accurately predict whether a patient will survive for 5 years or longer or will succumb to the disease within 5 years.

Methodology

I will train a neural network using the available data, and then evaluate its performance by testing it on unseen data.

Below are the detailed steps of the project in order.

Data Preprocessing: This involves importing the data, visualizing to get a better understanding of how we can use it more efficiently, and normalizing and scaling the input variables (with MinMaxScaler from scikit-learn) to facilitate effective training of the neural network.

Visualizing the data: Figure 1 shows the relationship between Age and the number of positive axillary nodes. From Figure 1 we can see that the higher the number of positive axillary nodes, the more the probability of not surviving. Moreover, we can conclude that age is not proportional to the number of positive axillary nodes.

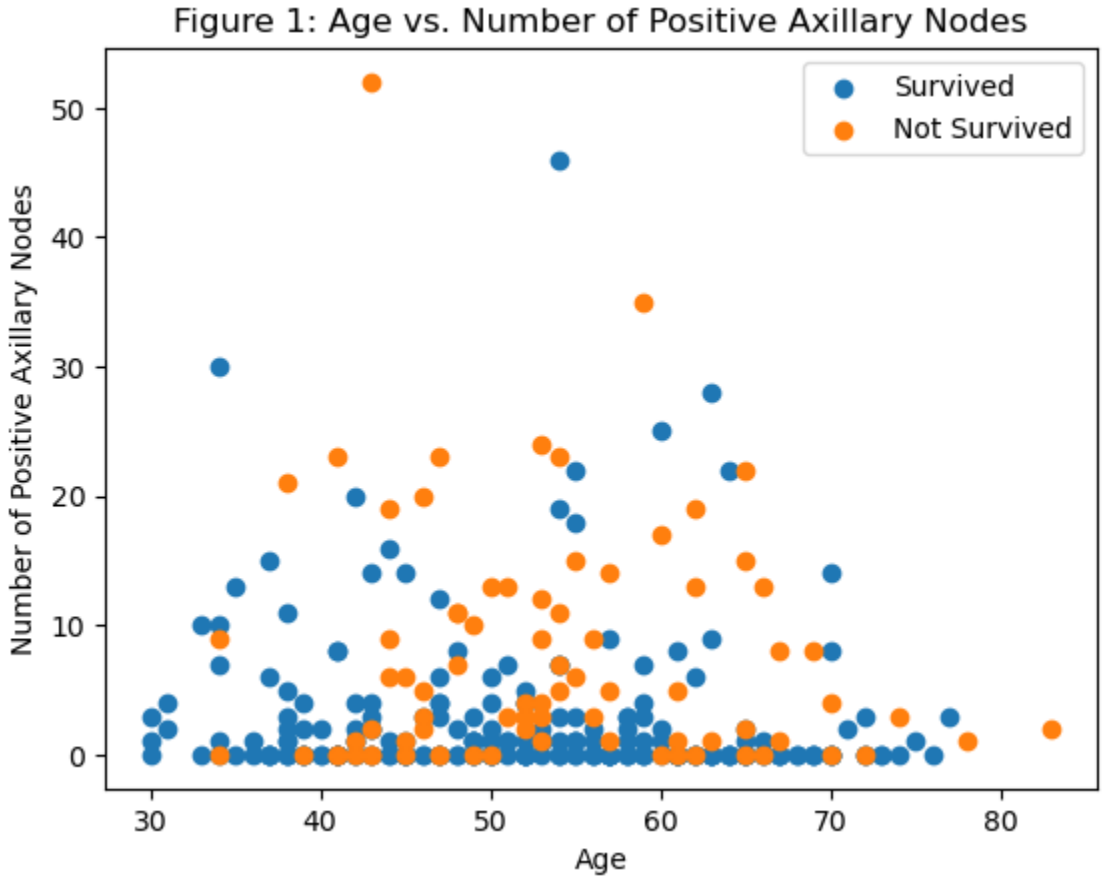


Figure 2 shows the number of cases of survival and death by age. From Figure 2 we can see that after some age threshold (approx 75), 100% of the patients don't survive more than 5 years, and, in contrast, young people (< 35), live for more than five years after the operation.

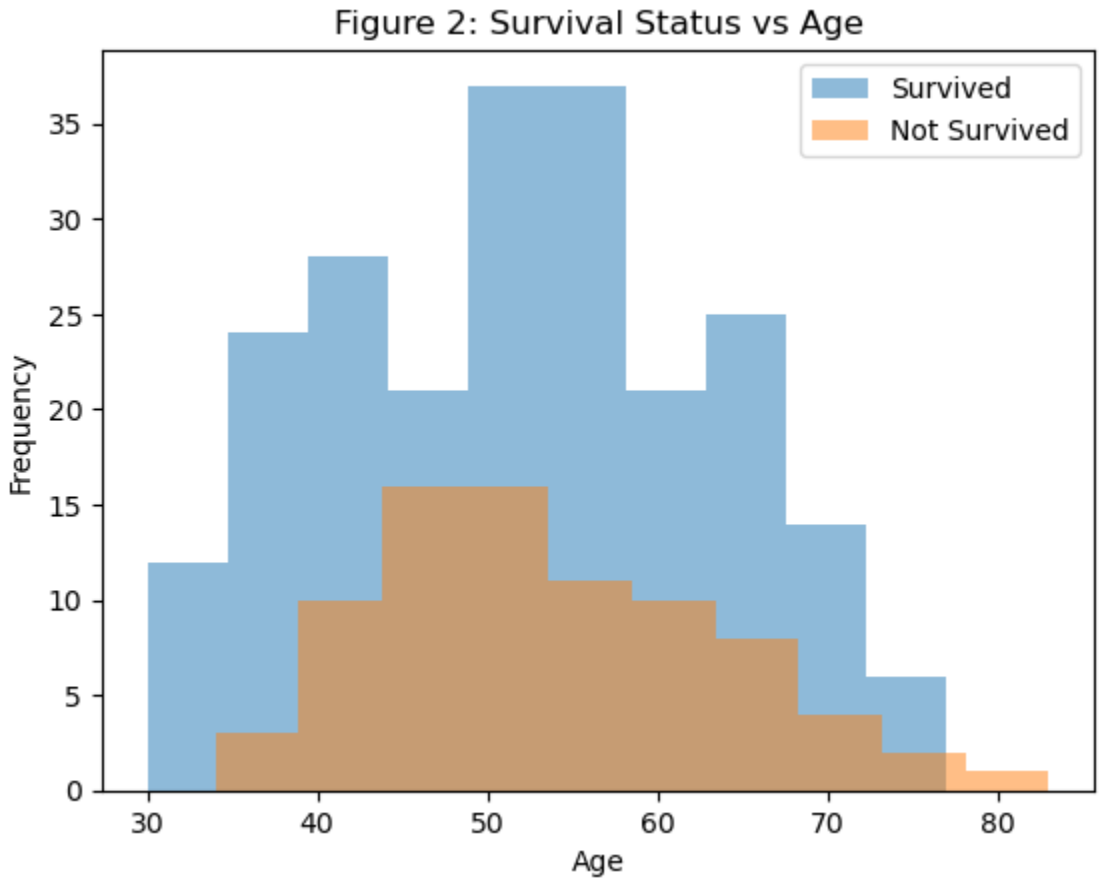
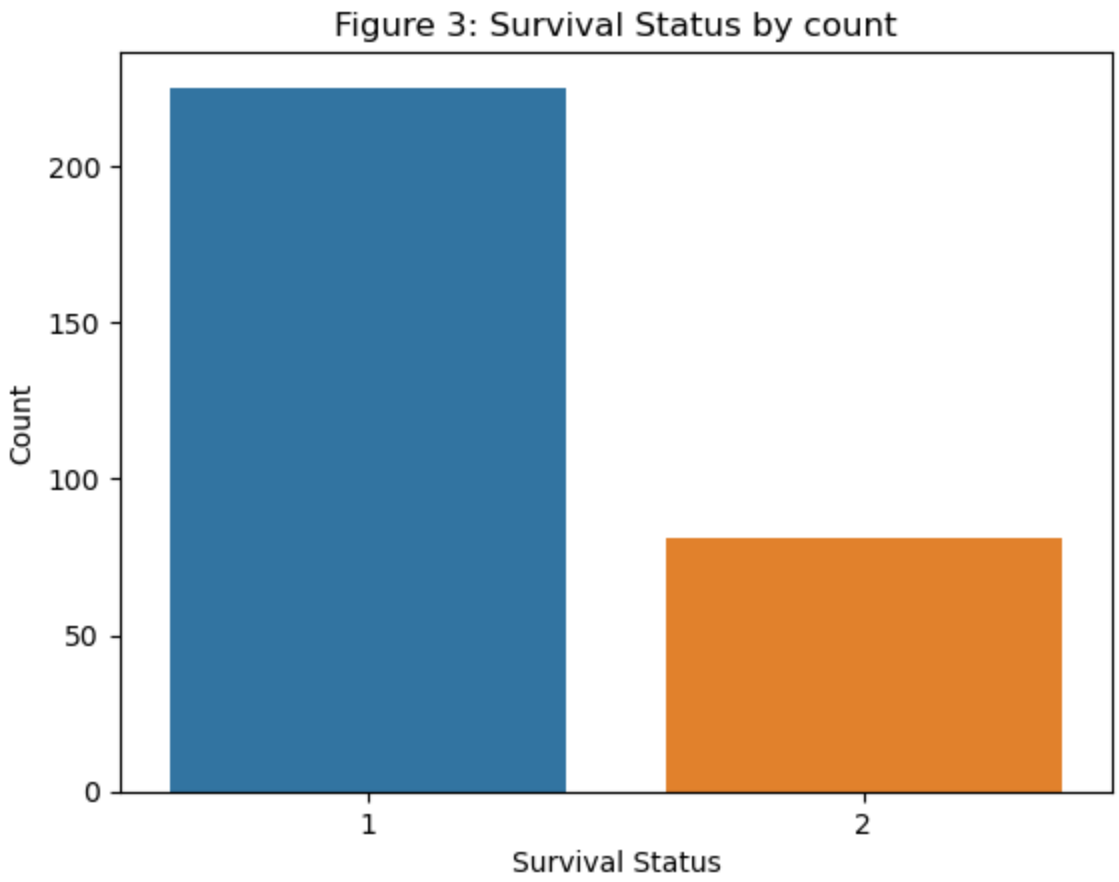


Figure 3 simply shows the number of cases in each of our two categories (surviving for more than 5 years, and living for less than 5 years after the operation). From Figure 3 we can conclude that the majority of the patients in the database lived more than 5 years after the operation.



In summary, we do not have any outliers or missing or noisy data, therefore, no further need for processing the data. In this step we also divide the data into a test and train set (70% train and 30% test), and normalize the data with MinMaxScaler.

Supervised learning

Designing the architecture of the neural network involves determining the number of layers, the number of neurons in each layer, and the activation functions to be used. Moreover, we changed the learning rate and the number of epochs. The processed dataset was divided into training and validation sets. The neural networks are trained on the training set, and the weights and biases are adjusted iteratively using optimization algorithms. After the network is trained, it is evaluated on the validation set to assess its performance and generalization ability. We simulate the

network on the test set, get the outputs, and compare them to the actual output values. Afterwards, the accuracy score is calculated for the network.

All in all, I have ten different neural network architectures (all feed forward neural networks). I will explain each of them very briefly and give the accuracies for each:

1) 1 hidden layer with 6 nodes, 3000 epochs, learning rate: 0.3, 30/70 split, accuracy: 71.739130
2) 1 hidden layer with 3 nodes, 3000 epochs, learning rate: 0.3, 30/70 split, accuracy: 67.391304
3) 3 hidden layer with 6 nodes, 3000 epochs, learning rate: 0.3, 30/70 split, accuracy: 68.478261
4) 2 hidden layer with 10 nodes, 3000 epochs, learning rate: 0.3, 30/70 split, accuracy: 67.391304
5) 1 hidden layer with 6 nodes, 3000 epochs, learning rate: 0.001, 30/70 split, accuracy: 67.391304
6) 1 hidden layer with 6 nodes, 10000 epochs, learning rate: 0.3, 30/70 split, accuracy: 68.478261
7) 1 hidden layer with 6 nodes, 3000 epochs, learning rate: 0.3, 50/50 split, accuracy: 71.241830
8) 1 hidden layer with 6 nodes, 3000 epochs, learning rate: 0.3, 60/40 split, accuracy: 71.544715
9) 1 hidden layer with 6 nodes, 3000 epochs, learning rate: 0.3, 80/20 split, accuracy: 58.064516
10) 1 hidden layer with 6 nodes, 3000 epochs, learning rate: 0.3, 90/10 split, accuracy: 61.290323

Based on the accuracy scores for the 10 different neural networks with varying parameters, we can draw the following conclusions:

- Networks with 1 hidden layer consistently achieved higher accuracies compared to networks with 2 or 3 hidden layers. Among the networks with 1 hidden layer, the best accuracy of 71.739130% was obtained with 6 nodes in the hidden layer. Networks with 3 nodes in the hidden layer yielded lower accuracies compared to networks with 6 nodes.

- Increasing the number of epochs from 3000 to 10000 did not result in a significant improvement in accuracy. This suggests that the networks do not learn much after a certain number of epochs.

- The learning rate of 0.3 consistently provided better accuracy compared to the learning rate of 0.001. A higher learning rate allows the network to adjust its weights more quickly during training, potentially leading to better convergence.

- Networks trained with a 30/70 or 50/50 split achieved higher accuracies compared to other split ratios. However, with a 50/50 split we have risks of too much generalization. Networks trained with an 80/20 split or a 90/10 split had notably lower accuracies. Bigger train set may result in overfitting the data. Larger training sets (higher split ratios) tend to provide more representative data for learning and generalization.

The network with 1 hidden layer, 6 nodes, 3000 epochs, a learning rate of 0.3, and a 30/70 split ratio consistently achieved the highest accuracy of approximately 72%. This network has a good balance between model complexity and training efficiency.

In conclusion, Network number 1 is trained and validated, it can be used to make predictions on new, unseen data to determine the survival status of breast cancer patients after an operation with 72% accuracy. Further changes can be done to achieve more accuracy, but because of limited time, I did not employ further changes to the algorithm.

Unsupervised Learning

For unsupervised learning algorithms, we give the network the processed data without the classes. First, we determine the optimal number of clusters by the elbow method and the silhouette score. The elbow method gives as a result a plot which shows how the error changes proportional to the number of clusters. We can see from Figure 4 that 4 clusters is the most efficient, since after that when we increase the number of clusters the error does not change much, but the complexity of the algorithm increases. So with the elbow method we choose 4 as the optimal number of clusters. With Silhouette Score, the higher the score, the better the accuracy of the algorithm. We can see from Figure 5 that 3 clusters give the highest Silhouette score.

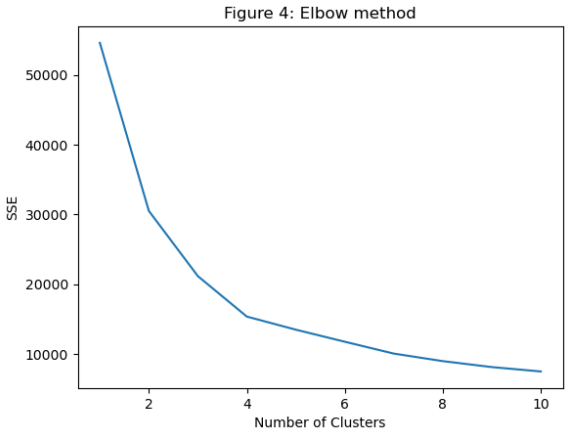


Figure 4: Elbow method

Figure 5: Silhouette Score

We create a Self organizing Map (or a Kohonen Network) with the method newc from the neurolab library. We specify the minimum and maximum values for each bit of our data, and the number of clusters. I created two separate networks, one with 4 clusters, another one with 3 to see which one gives a better result.We can see the results of the two networks in the corresponding figures.

Figure 6: Kohonen network with 4 clusters

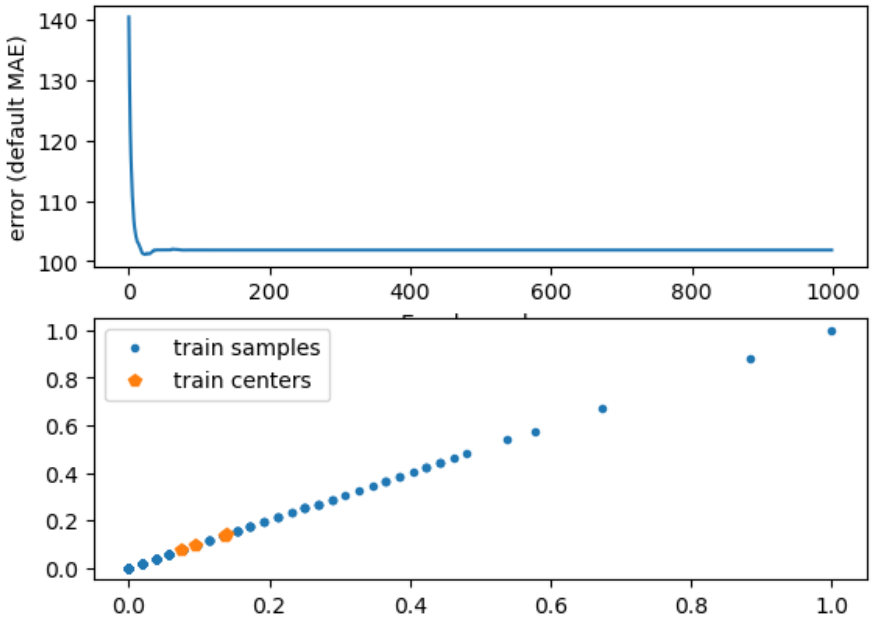
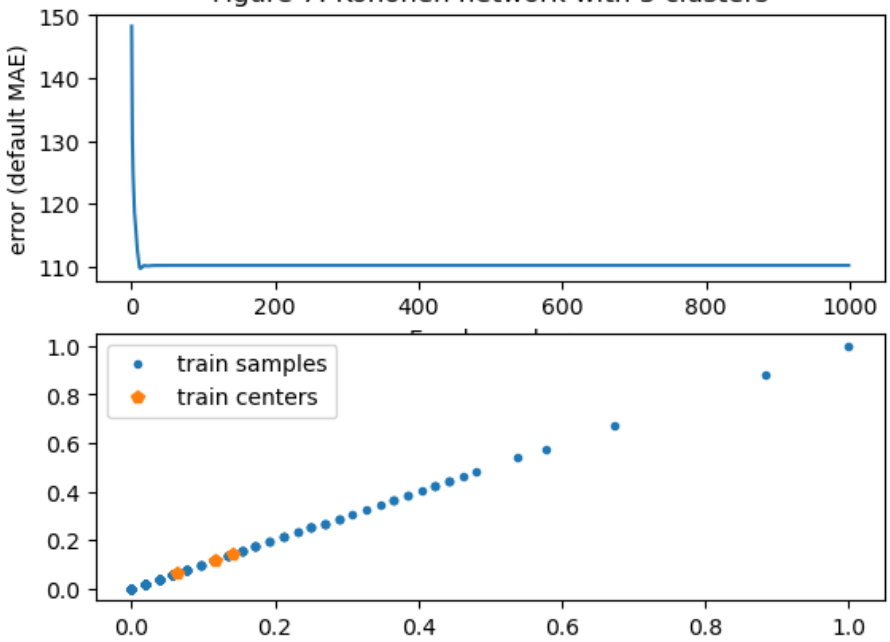


Figure 7: Kohonen network with 3 clusters



To summarize, with the network which clusters the data around three clusters we can reach a lower error value (can be seen in Figure 6). Further changes can be done to decrease the error.

In summary, building a neural network involves important steps to make it work well. These steps include preparing the data, deciding how the network should look, training it, and checking how accurate it is. We also need to choose the best settings for things like how fast the network learns and how many times it should practice.

However, in my opinion the most important thing to have in mind when designing a supervised neural network is the number of output nodes and the number of hidden layers. For that, we need to carefully analyze and understand the data to find the best way to represent the input and output. It is also very important to pay attention to the split ratio, not to overfit or too much generalize the model. The most important step when designing an unsupervised neural network model is to determine the optimal number of clusters.