Using Artificial Neural Networks to Predict One Year Population Mortality Rates

SHUN HUANG
Using Artificial Neural Networks to Predict One Year Population Mortality Rates

SHUN HUANG

Master in Computer Science
Date: June 14, 2019
Supervisor: Erik Fransén
Examiner: Joakim Gustafsson
School of Electrical Engineering and Computer Science
Host company: Itello AB
Swedish title: Tillämpning av artificiella neurala nätverk för att förutspå ettårsdödlighet
Abstract

Being able to predict mortality rates is the key factor in any pension or life insurance companies’ business model. Artificial Neural Networks are already being tested and implemented to predict mortality in the field of medical science, with recent studies showing promising results of their predictive power in one year mortality rates. Today, insurance companies in Sweden utilizes the Makeham curve to model and approximate mortality, traditionally with only age and sex being its input features. This study utilized artificial neural networks to model one year mortality rates that could otherwise be derived from the Makeham curve. Features other than sex and age were also included as a part of this study to introduce more features that could affect mortality rate. The network was successful at modelling the one year mortality rates and it was able to distinguish between age, sex and the newly introduced features. It yielded results that were on par with predictions made by the Swedish branch organization of the private insurance companies.
Sammanfattning

Contents

1 Introduction 1
  1.1 Research Question ........................................... 2
  1.2 Objective ...................................................... 3
  1.3 Academic Impact ............................................. 3
  1.4 Societal Impact ............................................... 3
  1.5 Ethics Statement ............................................. 3

2 Background 5
  2.1 Mortality and Pension in Sweden ............................ 5
    2.1.1 Gompertz-Makeham law of mortality .................... 5
  2.2 Machine Learning and Mortality ............................ 8
    2.2.1 Literature Review ........................................ 8
    2.2.2 Multi-Layer Perceptron .................................. 14
    2.2.3 Grid Search .............................................. 19
    2.2.4 K-fold Cross Validation ................................. 20
    2.2.5 Testing and Evaluation Metrics ......................... 20

3 Methods 22
  3.1 Data ........................................................... 22
    3.1.1 Data Limitations ......................................... 22
    3.1.2 Data Cleansing .......................................... 22
    3.1.3 Feature Scaling ......................................... 24
  3.2 Training the Multilayer Perceptron ......................... 25
  3.3 Graphing the Results ........................................ 26

4 Results 27
  4.1 Grid Search .................................................. 27
    4.1.1 Layer and Nodes ......................................... 27
    4.1.2 Optimizer ............................................... 28
    4.1.3 Learning Rate ............................................ 28
4.1.4 Batch Size and Epochs ................. 28
4.2 Final Model .................................. 29
   4.2.1 Performance ........................... 29
   4.2.2 Entire Population Group .............. 30
   4.2.3 Population Group Divided into Sex .... 31
   4.2.4 Population Group Divided into Industry ... 32
   4.2.5 Prediction Comparison .................. 35

5 Discussion ...................................... 36
   5.1 Methodology Limitations and Decisions ...... 36
   5.2 Network Performance and Meta-Analysis .......... 37
   5.3 Future Studies ............................. 38

6 Conclusions .................................... 39

Bibliography .................................... 40
Chapter 1

Introduction

Death, is the one certain factor across human history. What is uncertain is how long one will live before dying. In general, the life expectancy of population groups throughout history have all been mapped out. It is clear that average life expectancy of populations is on a steady rise \[1\]. Evidence shows that this is due to better medicine, hygiene and general quality of life in society, that improves the overall health of a population.

The uncertainty stems from causes of death in science. Cause of death, recorded on official death certificates is divided into two categories "Natural Death" and "Unnatural Death", while there is no international standardized definition \[2\], a simple generalization will be made for this study to introduce death and its relationship to insurances and pension.

- Natural Death is defined as death occurring in the course of nature and from natural causes (such as age or diseases) \[3\]. Life Expectancy is usually calculated through natural death within any period of time, unless there is a major event known to trigger a significant proportion of unnatural deaths, such as World War 2.

- Unnatural Death is defined as death that cannot be described as Natural Death, and usually unpredictable \[3\]. It includes, but not limited to: accidents, war, murders and suicides.

Pensions stem from natural death, associated with life expectancy and aging. Due to aging consequences, individuals are unable to work but still need an income to be a functioning member of society. Thus pension, also known as benefit plans have been laid out, where money is invested pre-retirement and can be retrieved post-retirement. The first universal public pension system was laid out by Sweden in 1913 \[4\].
Pension – an amount of money paid regularly by the government or a private company to a person that has become too old and is classified as retired. Prices and premiums need to be calculated based on the average populations’ life expectancy. According to the Swedish Pension Authority in 2016 [5], pension capital invested in funds in Sweden was valued at approximately 4950 billion SEK. In Sweden there are 3 primary types of pension [5]:

- National pension “Allmän pension”, administered by the government.
- Occupational pension “Tjänstepension/Avtalspension”, administered by a company where a part of the salary is paid into a pension.
- Private pension “Private pension”, administered by the individual who pays a monthly premium to an insurance/pension company

The fundamental assumption to all pensions is that each individual cannot live indefinitely. For collective agreements, the salary being paid into the pension is pooled. This pool of capital in this fund is then invested in various ways. For money to be paid out from this pool, one needs to be declared retired. In a perfect scenario, the remaining capital in this pool reaches 0 SEK once the last person in the group that constitutes this pool is deceased. This is why the study of mortality is imperative for the business model of a company providing pension management services.

Ultimately, if one were able to be accurately predict the population mortality, then companies would be able to offer pension products and services that are very attractive and fair to its customers, while still being profitable for the company.

1.1 Research Question

To what extent can artificial neural networks be used to predict one year mortality probabilities? Can it outperform the traditional mathematical models used? A group in this context is defined as a population that can be divided into sets based on discernible factors, such as sex, income, field of work etc. If possible, can one incorporate more dimensions than the ones currently used today to obtain a more accurate model of mortality in groups?
1.2 Objective

The objective of this report is to create a model and compare to the current predictions employed by insurance companies in Sweden, and further divide population groups into more distinct groups that provides a more accurate representation of mortality.

1.3 Academic Impact

While mortality studies using artificial neural networks have been done before, it has been mainly studied in the context of health care services. These studies, which will later be covered, are mainly used to assess a patient’s chance of survival within a one/two year period due to different medical complications, and the model is then benchmarked to traditional assessment methods [6][7]. Regarding the combination of insurance and mortality studies using artificial neural networks, the Lee Carter model has been studied, however it is to this author’s best knowledge that predicting the one year mortality probabilities of a population that is mathematically approximated from the Makeham model has not been been done before.

1.4 Societal Impact

It is required by law to be as transparent as possible in the insurance industry, showing all calculations that are done to arrive at the business model when setting prices, fees, due payments and payouts. Due to the nature of Machine Learning, being a “black box”, it is not feasible to set pricing and such on pensions/insurances directly based on the results from the output of a machine learning model. However, any valid results from the machine learning model can indicate that calculations may be off or correct, being a benchmark to the mathematicians that work with death calculations.

1.5 Ethics Statement

- All individuals in the database are unidentified before use, and are accessible by all developers at Itello. National identification number is removed. Address in both city and street number level are randomly assigned to new values and cannot be traced back.
• This study does not present or publish any individual information that may identify a person, directly or indirectly. Only fictional examples are presented.

• This study does not publish any of the raw data used. Only general statistics of this data-set is published. Only fictional examples of what this data-set may represent are presented.
Chapter 2

Background

2.1 Mortality and Pension in Sweden

2.1.1 Gompertz-Makeham law of mortality

Note: This subsection is intended for the readers that are unfamiliar with one of the mathematical models used today by insurance companies in Sweden for death rate calculations, in which this study attempts to model based on an unseen population.

Mortality is a hazard rate. Hazard rate is the rate of death for an item given a certain age. The hazard rate is applied to items that cannot be repaired.

The hazard rate is determined from the general hazard equation:

\[ h(t) = \frac{f(t)}{R(t)} \] (2.1)

where \( f(t) \) is the probability density function, in a hazard rate it is the time interval in which the death of the item occurs. \( R(t) \) is the survival function, in this case, the probability that the given item survives past the time interval. In the context of this study, the item is a human, and the hazard rate is applicable since a deceased human cannot be repaired or revived.

During the 1930’s, in an attempt to model mortality distribution, Swedish authorities decided to apply Gompertz-Makeham Law as the function to model the hazard function [1].

According to Gompertz-Makeham, the probability distribution function for the death rate can be modelled:
PDF\( (x) = (\alpha \ast e^{\beta x} + \lambda) \ast \exp(-\lambda x - \frac{\alpha}{\beta}(e^{\beta x} - 1)) \) (2.2)

This can be directly translated to the \( f(t) \) component of the hazard function.

The cumulative distribution function for death rate is:

\[ CDF(x) = 1 - \exp(-\lambda x - \frac{\alpha}{\beta}(e^{\beta x} - 1)) \] (2.3)

Meaning that the survival function is \( 1 - CDF(x) \), which then can be directly translated to the \( R(t) \) function of the hazard function:

\[ R(t) = 1 - (1 - \exp(-\lambda x - \frac{\alpha}{\beta}(e^{\beta x} - 1))) \]
\[ = \exp(-\lambda x - \frac{\alpha}{\beta}(e^{\beta x} - 1)) \] (2.4)

Thus the hazard function for human mortality can be found:

\[ h(t) = \frac{(\alpha \ast e^{\beta x} + \lambda) \ast \exp(-\lambda x - \frac{\alpha}{\beta}(e^{\beta x} - 1))}{\exp(-\lambda x - \frac{\alpha}{\beta}(e^{\beta x} - 1))} \]
\[ = \alpha \ast e^{\beta x} + \lambda \] (2.5)

Since the rest of this subsection will be primarily referring to a report of insurance and death by the Sweden insurance authority, we will be translating the hazard function into the the formula recognized by the report. This only involves renaming the variables:

\[ h(t) \rightarrow \mu_x \]
\[ \alpha \rightarrow \beta \]
\[ \beta \rightarrow \gamma \]
\[ \lambda \rightarrow \alpha \] (2.6)

And thus we arrive at the formula used by the report:

\[ \mu_x = \beta \ast e^{\gamma x} + \alpha \] (2.7)

Where \( \mu_x \) is mortality intensity given current age \( x \), note that \( x \) is continuous in this formula and not distinct. \( \alpha \) is a constant variable, represents the risks that result in mortality for non-age dependent factors. \( \beta \) is a coefficient to adjust the Makeham model properly. \( \gamma \) is the age dependent variable, representing the risks that result in mortality for age dependent factors.
\( \mu_x \) can thus be interpreted as the probability of death in any moment for a person based on their age.

In the end of the 1980’s, A massive study was carried out by G.Stoltz to investigate mortality and it concerned 17 of the Swedish Insurance Companies. This resulted in the famous M90 mortality data-set with its tables.

For the M90 data-set, the Makeham curve was modified:

\[
\mu_x = \beta \times 10^{\gamma(x-f)} + \alpha 
\]  

(2.8)

According to the M90 data-set, \( \alpha = 0, 001 \), \( \beta = 0, 000012 \) and \( \gamma = 0, 044 \) were all retroactively set. Instead of using \( e \) as the exponential factor in the Gompertz - Makeham formula, a exponent of base 10 was used. \( f \) acts as a translation in age difference between male and female which was set at \( f = 6 \), so two curves would be produced [1].

In a new law set in 2011 by the European Union [8], insurance companies were no longer allowed to discriminate the pricing of their insurances based on sex. This meant that the \( f \) variable is today made redundant.

The connection to one year mortality probabilities

By combining Makeham’s formula and M90, the mortality curve could be obtained, and one could estimate the average life expectancy remaining from a certain year.

Example according to the M90 data-set:

- If a female was 50 years old, she would be expected to live another 39.4 years – for a total life expectancy of 89.4 years.
- If a female was 65 years old, she would be expected to live another 25.6 years – for a total life expectancy of 90.6 years.

Average remaining expected life can be expressed as:

\[
T_x(t) = \text{remaining time to live for an } x \text{ year old individual at time } t 
\]  

(2.9)

What this study covers is the ability to predict the probability of an \( x \) year old individual deceasing within a year, denoted as \( q_x \), this will be referred to as the 1-year mortality probability. Mathematically, this can be expressed as:
The relationship between $q_x$ and $u_x$ for any one year period is:

$$q_x = 1 - e^{\exp\left(-\int_x^{x+1} \mu_s ds\right)}$$  \hspace{1cm} (2.11)$$

But because discrete values will be used, $q_x$ needs to be approximated, an explanation of the approximation is explained in Försäkringsföreningen [1]’s report.

Thus, the equation that this study will evaluate against is:

$$q_x \approx \frac{\mu_x \frac{1}{2} }{ 1 + \frac{\mu_x \frac{1}{2}}{2} }$$  \hspace{1cm} (2.12)$$

In the report by Försäkringsföreningen [1], it is mentioned $q_x$ is approximated through MacLaurin series expansion, and can be approximated better if expanded to a higher degree.

\section{2.2 Machine Learning and Mortality}

\subsection{2.2.1 Literature Review}

\textbf{L. Shi et al. on Mortality and ANN}

Shi, Wang, and Wang [6] did a study on predicting 1-year mortality in elderly patients with a medical condition known as intertrochanteric fractures in China. For this prediction, the neural network used was a multi-layer perceptron.
A stepwise selection of patients was used to refine data from the database that was not relevant. For example, everyone under 65 years of age and the fractures that were not intertrochanteric were first removed. Then the patients ineligible for surgery were removed, as the risk of death was only relevant for those who underwent surgery. The data-set was divided into 66.6% training data, and 33.4% testing data. A hold-out data-set was not used.

Medical conditions such as diabetes, cancer etc were chosen as relevant features for the network. According to the authors, these factors are established as factors that influence outcomes after intertrochanteric fracture surgery. For many features, it was a binary class, such as a yes/no answer to whether patients had Diabetes etc. However, age and mobility score were also grouped respectively so they could be set as a binary class, according to the authors, it was due to them wanting to rapidly apply the predictive models clinically. Another limitation described was the database itself as it was limited to only a few thousand sets.
The network setup was fairly standard, it consisted of an input layer, hidden layer and output layer. It appears that the authors of this paper did a manual grid search for 10, 15 and 20 neurons in the hidden layer. The computer was also programmed to automatically select a number of neurons in the hidden layer. 20 multi-layer perceptrons were trained simultaneously, and the results from the four most accurate networks were averaged to obtain the final result.

This was a similar study that attempted to predict 1 year mortality rates. Shi, Wang, and Wang [6]'s step-wise selection of data is a method this study will be using. The motivation of why the authors chose certain features will also taken into consideration, for example, there are other studies where life expectancy is shown to be higher in high income areas [9].

The methodology of optimizing the network is limited. For example, for age and new mobility score, the authors could have used a more standard approach of feature scaling method such as normalizing through min-max method or standardization. Also, grid search should be more comprehensive, not only limited to selecting the number of nodes in the hidden layer. The other critical assessment of this study is that the data set was limited in terms of machine learning, and there was thus the increased risk of over-fitting, a sample size this small could also imply that there was another factor that constantly reduced or increased the risk of mortality, which one wouldn’t find in another data set. Also due to their lack of appropriate feature scaling methods, combined with the small data set, there’s a high probability that outliers here...
will impact the model too much, making it less robust.

It was also unclear how this study minimized the loss function. No men-
tions of the SGD variants and more popular optimizers used today, and other
factors such as momentum etc. There was also no mention of dropout, batch
normalization and such. All in all, this paper is credible (published in peer
reviewed Brazilian Journal of Medical and Biological Research), this paper
proved that linear regression could be beaten.

**Monsalve-Torra et. Al. on Mortality**

Monsalve-Torra et. Al. (2016) did a study on mortality prediction within pa-
tients that undergoes a specific surgery called open repair of abdominal aor-
tic aneurysm. For these predictions, three different networks were used. A
multilayer perceptron, a radial basis function neural network and a bayesian
network.

![Radial basis function neural network](image)

**Figure 2.3:** An example of Radial basis function neural network used with its
respective functions (Monsalve-Torra et Al., 2016, p197).

In similar fashion to Shi, Wang, and Wang [6]’s work, the features used
were assessed to be relevant for the networks. Like the previous paper, the
data processing was similar, ages were grouped, with no mention of whether
each feature would be better off one hot encoded. It was also uncertain if
they purposely grouped ages or other features together in what is known as
bucketization, which can be beneficial if for example, data-sets were sparsely
represented, however this is still unclear. Difference here is the detail of the
report. K-fold cross-validation technique was mentioned for the training. Af-
fter the first stage, a feature selection process was applied to reduce computa-
tional complexity, over-fitting and improve generalization. The hyper parameters were apparently empirically found.

Instead of only comparing accuracy, the performance measuring methods: sensitivity:

$$\frac{TP}{TP + FN}$$

(2.13)

and specificity:

$$\frac{TN}{FP + TN}$$

(2.14)

where

$$TP = True \ Positives,$$

$$FN = False \ Negatives$$

$$TN = True \ Negatives$$

$$FP = False \ Positives$$

(2.15)

were also included. This is because while accuracy was high (>90%), sensitivity could still be very low (<60%), as indicated in their study. The context is that the hospital needs to plan better care to patients with high risks thus if sensitivity is low, it means that there will be a lot of patients that won’t get the care they need because they were classified as false negatives.

All in all, from Monsalve-Torra et al. [7]’s work, there are a lot of applicable theory in machine learning that will be used. The paper is also credible, as it has been published in a peer-reviewed journal known as “Journal of Biomedical Informatics”. It also clear that multi-layer perceptrons is suited for these kind of studies.

Data

According to Statistics Sweden [10], there is a standard for grouping occupations abbreviated SSYK. This standard is made up of 10 categories, in which these categories can be further divided into 3 more hierarchies, which translates to a total of 429 different occupations. Also, every occupation is classified into a qualification level:

- No education requirement beyond primary school
- Gymnasial education requirement or equivalent
Shorter University/College education, < 3 years

Longer University/College education, > 4 years

There is also the group of people that are unemployed, but this won’t be accounted for, as this study delimits to the data that extracted off of people who have an insurance through their employer. It is also implausible to use 429 one-hot-encodings for occupations, as data will be then very sparsely represented. Dividing into into either the 10 categories or 4 qualification levels is more appropriate when represented in one hot encoding.

In another report by Statistics Sweden [11], it was mentioned that in general, there is a clear relationship between the mortality and the occupations working environment. However there is also an adequate relationship between salary and mortality, but there are many exceptions [11]; Low salary and low mortality exist in occupations dominated by females, examples being: teaching in all levels, pedagogues, dental hygienists, and receptionists. The opposite is true as well, high salary and high mortality exist in occupations that are over-represented by males, examples being pilots, ship commanders, well drillers. Thus, salary, occupation and sex will be used.

These occupation levels are not available Itello’s database, however there is data on the type of pension insurance each individual has. The following four pension programs exists in the data set that this study will be using:

1. PA16 pension is represented by government employees such as: police, military, university teacher, air traffic controller etc. [12]

2. SAFLO pension is represented by private workers such as: construction workers, hotel employee, restaurant employee, factory worker etc. [12]

3. KAPKL pension is represented by communal and city council employees such as: teachers, firefighters, doctors, librarians etc. [12]

4. ITP pension is represented by private employees such as engineers, journalists, economist and lawyers etc. [12]

It should also be mentioned that the data set does not contain salary, but it does contain how much money is paid into fund of the individual by their company every month, and thus one can deduce their actual salaries. Therefore, by including the payments as salary and pension type as industry the network should be able to learn, for example, if someone in the SAFLO pension is a construction worker, or an upper-level manager.
Another motivation of using pension types is that they also exhibit non-linear relationships with other features and the target. For example, both female workers and male workers exhibit high mortality in machine operation for bleaching, dyeing and washing [11]. While the occupation with highest mortality among exclusively females is tele/electronic reparation, and for males respectively is machine engineer [11]. For the machine engineers in the second level occupation classification, 45% had a post-gymnasium education, while 55% didn’t. For machine operation in the second level occupation classification, only 9% had a post-gymnasial education.

If one were to make a very simple and over generalized hypothesis:

<table>
<thead>
<tr>
<th>Occupation</th>
<th>Gender</th>
<th>Resulting Mortality Risk</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tele/Electronic Reparation</td>
<td>Male</td>
<td>Low</td>
</tr>
<tr>
<td>Tele/Electronic Reparation</td>
<td>Female</td>
<td>High</td>
</tr>
<tr>
<td>Machine Engineer</td>
<td>Male</td>
<td>High</td>
</tr>
<tr>
<td>Machine Engineer</td>
<td>Female</td>
<td>Low</td>
</tr>
</tbody>
</table>

Table 2.1: This table represents the hypothesis.

Assuming that there’s only two occupations and genders, then in a binary classification it could be represented as:

<table>
<thead>
<tr>
<th>Occupation</th>
<th>Gender</th>
<th>Resulting Mortality Risk</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

Table 2.2: Conversion of table 2.1 into a binary representation.

Table 2.2 also models an XOr gate. XOr inputs are not linearly separable and it’s a classic introductory problem to multi-layer perceptrons and an example that can be solved by adding a hidden layer in a perceptron network [13].

2.2.2 Multi-Layer Perceptron

This sub-section is intended for the readers that are unfamiliar with Multi-Layer Perceptrons and some methods and techniques of training and evaluating a network. This section will also explain why it should be theoretically
possible to predict and model the one year mortality probability curve given by \( q_x \approx \mu \frac{\mu + 1}{1 + \frac{\mu + 1}{2}} \) that was previously introduced. This is also important for the readers who wishes to understand the grid-search section of the results.

The Perceptron

A multilayer perceptron is a network of neurons, which are called perceptrons. Perceptrons were introduced in 1958 by Rosenblatt [14]. It was an electronic device which was designed based on the principles of a biological neuron, and it showed an ability to learn. In the context of machine learning, the perceptron is a node that can compute a single output given multiple inputs, known as input nodes.

This is done by forming a linear combination of the inputs with weights that are associated with each input, when passed through an activation function, will result in an output. The design is based upon the biological principle of the action potential of neurons. Where the linear combination of inputs and weights is the stimulus. Depending on the strength of the stimulus, it may trigger the action potential of the nerve, which is represented by the activation function.

Mathematically, a single perceptron with its inputs and output, can be modelled as such [15]:

\[
y = \varphi \left( \sum_{i=1}^{n} w_i x_i + b \right) = \varphi(W^T X + b) \tag{2.16}
\]

where

\[
y = \text{the output} \\
\varphi = \text{activation function} \\
W = \text{the vector of weights} \\
X = \text{the vector of inputs} \\
b = \text{the bias} \tag{2.17}
\]

It can be visually represented as the following:
The Layers

The limitation of a single layer perceptron is that it is not able to learn non-linear functions, an example such as the XOR gate. In 1969 Minsky and Papert [16] proposed multiple layers, however the original linear problem could still not be solved until the implementation of the back-propagation algorithm into perceptrons published by Rumelhart, Hinton, Williams, et al. [17] in 1986.

The source nodes form the input layer. This layer is then connected to the first hidden layer, which consists of one or more perceptrons. The output of each of these perceptrons act as the input to the next layer, which can either be another hidden layer if desired, or ultimately the output layer that outputs the desired answer of this network.

A visual example of a fully connected network with one hidden layer is presented by figure 2.2 on page 10.

This study will test different configurations of layers and nodes in each layer.

The Weights

Note that the weights vector can never be initialized to be the same for each perceptron, else the output of these perceptrons will be the same, making all but one redundant. Thus, random initialization of weights is the normal procedure.
The Activation Function

The activation function of a node defines what it outputs. There are numerous activation functions that are both linear and nonlinear. If one were to look at the partial component of the perceptron:

$$W^T X + b$$  \hspace{1cm} (2.18)

We can see that it’s a linear function. In order to introduce non-linearity in an output, then a non-linear activation function is needed.

In this study, the ReLu \cite{18} and sigmoid functions are used in the final network.

The activation function employed between the input layer to the hidden layers and between the hidden layers is the rectifier (ReLu) function:

$$f(x) = x^+ = max(0, x)$$ \hspace{1cm} (2.19)

The advantages and drawbacks of using ReLu will not be explained in this paper, for further reading, please refer to Krizhevsky, Sutskever, and Hinton \cite{19}’s paper.

The activation function employed between the final hidden layer and the output layer is the sigmoid function:

$$f(x) = \frac{1}{1 + e^{-z}}$$ \hspace{1cm} (2.20)

The reason why a sigmoid function is employed as the final output activation is because $f(x) : [0, 1]$, meaning that it can output only values between 0 to 1. The consequence of the range is that its output can be interpreted as the probability of it belonging to class labelled 1.

For binary classification/prediction, the sigmoid activation function is always employed. A threshold is set on the Sigmoid activation function that classifies the output into either class 0 or 1. The threshold is usually set to 0.5 at default, meaning if $sigmoid(x) \geq 0.5$ then $y = 1$.

The Bias

As seen in the formula, the bias $b$ is usually employed to ensure that the network can still learn even when the linear combination of the input and the weight equals zero. In the case of ReLu, assuming that $W^T X$ is negative, assuming it can still make the input positive, allowing the output to be positive and giving it a positive gradient of 1, which will allow this node to learn.
The Loss and Cost functions

For binary classification, one can expect that the network will classify the input to belong to class A, while the correct label is class B. This type of loss has to measured as it is a way for the network to measure its performance for a single training example. For a sigmoid activation function, the binary cross-entropy loss function is used:

$$\text{Loss}(y, p) = - (y \log(p) + (1 - y) \log(1 - p))$$  \quad (2.21)

This function can be graphed if the true label = 1 to demonstrate the loss given the probability:

![Log Loss when true label = 1](image)

Figure 2.5: Graphing the value of the loss when the the correct label is 1, source:[20]

Looking at figure 2.5, one can see that it heavily penalizes a prediction that is confident and incorrectly classified ($\lim_{p \to 0} \text{loss} = \infty$ while $y = 1$ in this case).

In an unbalanced data-set such as the ones for one year population mortality rates where an overwhelming majority of the population will survive, heavy penalties should occur in a feature set belonging to an individual that has deceased that year.
The Learning

This part is theoretically heavy, and will only be simply explained. If a more thorough understanding is needed, then please refer to Rumelhart, Hinton, Williams, et al. [17]’s work.

The perfect model would have a total loss (known as the cost) of 0. Therefore, learning is essentially the process of minimizing the loss function. As the weights ultimately decide the output, the task at hand is to update the weights so that the loss produced is minimal.

To be able to do this, one would need to calculate how much each node is responsible to the overall contribution of the loss, which is calculating gradient of the error function with respect to the neural network’s weights. This is done through a process called back-propagation (Short for the backward propagation of errors, because the error (loss) is distributed backwards in the network after it’s calculated at the sigmoid node in this case).

The gradient thus reveals how a small change in each weight would affect the overall loss, and ultimately needs to be adjusted. The optimizer decides how these changes are to be made based on this gradient, and the learning rate decides on how much the weights are to be updated. This is known as gradient descent.

This study tests three different popular optimizers: RMSprop, Adagrad and Adam in a combination with different learning rates.

Gradient Descent is done iteratively to be able to converge to a minimal loss to where the weights are no longer able to update to reach an even lower loss. One update to the weights is one iteration.

A Batch-Size is the amount of feature vectors in the training data-set to be passed into the network to complete one iteration.

An Epoch is when the entire training data-set has been passed through the network forwards and backwards once.

Thus an epoch has been completed once all the batches have been iterated through.

This study will test different epochs and batch sizes.

2.2.3 Grid Search

In the previous section, there were many parameters in a network that could be affect the performance of the network. One way to estimate these parameters (known as hyper-parameters) is to make an educated guess of the interval of what the optimal values of each parameter is and test through these intervals.
Thus each different combination of given values constitutes the grid that will be tested.

### 2.2.4 K-fold Cross Validation

K-fold Cross Validation involves randomly splitting the data-set into k groups (known as folds) of similar size. Each fold will be held out once in which the model will be evaluated on, while the folds not being held out will be the training data-set that the model will be fit on [21]. This is done repeatedly until all folds have been held out once.

This is used to estimate the skill of model on unseen, and it’s ability to generalize answers. Each fold will yield it’s evaluation score, and the model with the best average score would be chosen.

The very broad pseudo code given by Brownlee [22]:

- Shuffle the data-set randomly
- Split the data-set into k groups
- For each unique group:
  - Take the group as a hold out or test data set
  - Take the remaining groups as a training data set
  - Fit a model on the training set and evaluate it on the test set
  - Retain the evaluation score and discard the model
- Summarize the skill of the model using the sample of model evaluation scores

### 2.2.5 Testing and Evaluation Metrics

Generally, accuracy is the most common method used to evaluate performance of a neural network. It is defined as the ratio of correct predictions to the total number of predictions made. This was the standard setup of the Keras skeleton code given.

\[
\text{Accuracy} = \frac{\text{Number of Correct Predictions}}{\text{Total Number of Predictions Made}} \tag{2.22}
\]

However, this should only be used when there is roughly an equal amount of samples in each class, and their weights are the same. In any modern population data set of a country, a very minuscule fraction the population is deceased and the rest are alive. For example, if 1% of a population is deceased while 99% of a population were alive. Then a dummy classifier that only outputs that the person will be alive would have an accuracy of 99%. The same case can be made for a trained classifier in logistic regression using the sigmoid activation function on the output layer with this study’s data-set, the default
cut-off threshold is at 0.5, implying that a person needs to have a predicted probability of 50% of dying to be classified as deceased.

Looking at the one year mortality rate curve [1], the average probability of a group of people dying only exceeds 50% at an age of >90, an age range that is not included in this study. This effectively means that if age were the only feature in the data set, everyone would still be classified as alive.

With the nature of pensions, it is also unimportant to classify whether individuals are alive or not, but rather predict the mortality rate of entire population groups, based on age, sex etc.

Due to the aforementioned reasons, accuracy will NOT be used. While the Area Under Receiver Operating Characteristic (AUROC or AUC) will be used.

The receiving operating characteristic curve is constructed by plotting the true positive rate (sensitivity) versus the false positive rate (fall-out). The area under this curve is thus called the AUROC. If the AUROC is 0.5, then it means the predictive power of this graph is completely random. If the AUROC is 1, then it means that the predictive power of this graph is perfect, and a perfect machine learning model has been achieved.
Chapter 3

Methods

3.1 Data

For the present study, a total of 3241847 customers were used and treated as individual data-points in the data-set. A total of 0.08% of this population data set are deceased as of 2018, while the remaining 99.92% are still alive.

3.1.1 Data Limitations

While several studies also show that health factors (diseases, habits and general vital functions) and geographic factors impact mortality, it was not possible to extract or induce this type of data from the database. Health factors did not exist in the database. Geographic data was non-linearly de-identified, meaning that populations from different cities and addresses could have grouped together into one makeshift city and address. Thus if one were to keep geographic data, it would be pure noise to this data-set.

This is due to the Inca software that Itello’s core business is built upon. Itello provides book-keeping of insurance and pension transactions in their system for insurance companies. Geographic data and health factors are very sensitive data and are only kept in the respective insurance companies’ databases, if any.

3.1.2 Data Cleansing

A significant amount of customers that did not fall into the premises of this study, and were systematically removed in the same manner as Shi, Wang, and Wang [6]’s study. This was done to guarantee that this study is to only be done
on statistics from the year of 2018 and removing any noise.

All Customers

OUT

Customers that deceased prior to 2018

Customers still alive and died during/after 2018

OUT

Customers that deceased in 2019

Customers still alive or died during 2018

OUT

Customers that does not hold an active contract

Customers still alive or death during 2018, with an active insurance, meaning, removing people that aren’t known to be either alive or deceased

Figure 3.1: The feature selection used, done in SQL

Example of a feature vector and it’s corresponding target vector:
<table>
<thead>
<tr>
<th>Data Point</th>
<th>Data Representation</th>
<th>Numerical Representation</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>$x_0$</td>
<td>Transaction Amount</td>
<td>$-\infty &lt; x &lt; \infty$</td>
<td>-1000</td>
</tr>
<tr>
<td></td>
<td>(SEK)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$x_1$</td>
<td>Gender</td>
<td>$\begin{cases} x = 0 &amp; \text{if female} \ x = 1 &amp; \text{if male} \end{cases}$</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>(female or male)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$x_2$</td>
<td>Age</td>
<td>$0 &lt; x &lt; \infty$</td>
<td>68</td>
</tr>
<tr>
<td></td>
<td>(year)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$x_3$</td>
<td>ITP</td>
<td>$\begin{cases} x = 0 &amp; \text{if not in industry} \ x = 1 &amp; \text{if in industry} \end{cases}$</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>(one hot encoded industry)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$x_4$</td>
<td>KAPKL</td>
<td>$\begin{cases} x = 0 &amp; \text{if not in industry} \ x = 1 &amp; \text{if in industry} \end{cases}$</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>(one hot encoded industry)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$x_5$</td>
<td>SAFLO</td>
<td>$\begin{cases} x = 0 &amp; \text{if not in industry} \ x = 1 &amp; \text{if in industry} \end{cases}$</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>(one hot encoded industry)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$x_6$</td>
<td>PA16</td>
<td>$\begin{cases} x = 0 &amp; \text{if not in industry} \ x = 1 &amp; \text{if in industry} \end{cases}$</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>(one hot encoded industry)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$y$</td>
<td>Death Status</td>
<td>$\begin{cases} x = 0 &amp; \text{if alive} \ x = 1 &amp; \text{if deceased} \end{cases}$</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>(alive or dead)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 3.1: An example of how a feature vector could look

Comment: Please note that the one hot encoded industries (ITP, KAPKL, SAFLO and PA16) are not industries themselves but are rather the names of the collective agreed pensions types that fall under different industries. This is already mentioned in the data section in the background.

It is possible that one person can have a pension from all four industries, this implies that this person has been or is currently employed at different employments in which the respective pensions types are offered.

### 3.1.3 Feature Scaling

All the features are standardized, meaning the features are rescaled so that the mean is 0 and the standard deviation from the mean is 1.

Looking at the standard formula for a gradient descent:

$$
\Delta w_j = -\eta \frac{\partial J}{\partial w_j} = \eta \sum_i (t^{(i)} - o^{(i)}) x_j^{(i)}
$$

(3.1)
Where

\[ w_j := w_j + \Delta w_j \]
\[ \eta = \text{learning rate} \]
\[ t = \text{the target} \]
\[ o = \text{predicted output} \]  

(3.2)

If the features were not standardized, it would imply that the weights would be updated much faster or slower using features that are not scaled down. For example, a common transaction value of 10000 will have about 100 times more impact of the weight update than a rare and extreme age of 100, this would effectively render the important information of age useless.

Thus the features need to be standardized, also referred to as Z-score normalization. First, the mean of each the class is calculated:

\[ \mu = \frac{1}{N} \sum_{i=1}^{N} (x_i) \]  

(3.3)

The standard deviation can then be found:

\[ \sigma = \sqrt{\frac{1}{N} \sum_{i=1}^{N} (x_i - \mu)^2} \]  

(3.4)

and thus the new value (also known as z-value) of each point in the class is calculated:

\[ z = \frac{x - \mu}{\sigma} \]  

(3.5)

The data-set used to calculated the standard deviation and mean is the training subset. During testing, this hold-out subset is also fitted according to the mean and standard deviation derived from the training subset. This is to ensure that the testing data is not misrepresented by another value as one would expect that the training and testing subsets have different means and standard deviations.

### 3.2 Training the Multilayer Perceptron

- The Multilayer Perceptron was set up using Keras. A code skeleton that was verified by the developers of Keras, was used as the basis for as-
sembling the Multilayer Perceptron. This was to ensure that the network functioned as expected.

- Cross validation and grid search is done through the GridSearchCV module from the Scikit-learn library that wrapped the MLP from Keras.

- 3-fold cross validation was chosen. The motivation of using 3 folds will be explained in the discussion.

- Grid search was done in the following order:

  1. Layers and nodes
  2. Optimizer with default learning rates
  3. Different learning rates from the optimizer chosen from previous step
  4. Batch size and epochs

  The motivation of why the grid search was done in this order will be explained in the discussion.

- The AUROC scoring function was chosen from GridSearchCV.

### 3.3 Graphing the Results

In the hold out set, each feature vector was assigned their predicted one year mortality rate, along with their true target value. This was then imported into Excel into pivot tables, where one can filter via the one hot encoded categories to view certain categories only. The average predicted one year mortality rates for each age was calculated as the mean of all predicted on year mortality rates in that given age group. The average true one year mortality rate for each age was calculated by summing the deceased population in divided by the entire population in the same age group.
Chapter 4

Results

4.1 Grid Search

In this section, for every configuration possible in the grid search, a 3 fold cross validation is done to find the average AUROC score of each configuration. Thus the presented answers were picked by the grid search as the best configurations through their average AUROC scores. For the complete data dump of all the average scores for each configuration, please refer to Appendix A.

4.1.1 Layer and Nodes

<table>
<thead>
<tr>
<th>Layers</th>
<th>1</th>
<th>2</th>
<th>3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Nodes</td>
<td>(28)</td>
<td>(28,28)</td>
<td>(28,3,28)</td>
</tr>
<tr>
<td>AUROC</td>
<td>0.86300</td>
<td>0.863949</td>
<td>0.862926</td>
</tr>
<tr>
<td>STD</td>
<td>0.002184</td>
<td>0.001350</td>
<td>0.001189</td>
</tr>
</tbody>
</table>

Table 4.1: Results of the layer and nodes grid search, presenting the best results picked by the grid search for amount of nodes and layers, with reference to each layer.

Comments: The (28,28) configuration had the highest area under receiver operating characteristic, and those two hyperparameters were thus chosen, although all configurations were still within a standard deviation of the (28,28) configuration.
4.1.2 Optimizer

<table>
<thead>
<tr>
<th>Optimizer</th>
<th>RMSprop</th>
<th>Adagrad</th>
<th>Adam</th>
</tr>
</thead>
<tbody>
<tr>
<td>AUROC</td>
<td>0.85445</td>
<td>0.861922</td>
<td>0.862245</td>
</tr>
<tr>
<td>STD</td>
<td>0.002627</td>
<td>0.001722</td>
<td>0.002192</td>
</tr>
</tbody>
</table>

Table 4.2: Results of the Optimizer Grid Search.

Comments: The Adam optimizer had the highest area under receiver operating characteristic, and was thus chosen as the optimizer for all layers except the output layer. Adagrad’s AUROC was also just one standard deviation away which meant that Adagrad could have performed better if the weight initialization was slightly different.

4.1.3 Learning Rate

<table>
<thead>
<tr>
<th>Learning Rate (Adam)</th>
<th>0.0001</th>
<th>0.001</th>
<th>0.01</th>
</tr>
</thead>
<tbody>
<tr>
<td>AUROC</td>
<td>0.863319</td>
<td>0.862519</td>
<td>0.862785</td>
</tr>
<tr>
<td>STD</td>
<td>0.002050</td>
<td>0.001789</td>
<td>0.001525</td>
</tr>
</tbody>
</table>

Table 4.3: Results of the learning rate Grid Search, these learning rates were specifically chosen for the ADAM optimizer.

Comments: The learning rate of 0.0001 performed best. However, similar to Table 4.1 all configurations were within a standard deviation of error.

4.1.4 Batch Size and Epochs

<table>
<thead>
<tr>
<th>Batch-size</th>
<th>64</th>
<th>128</th>
<th>256</th>
</tr>
</thead>
<tbody>
<tr>
<td>Epochs</td>
<td>10</td>
<td>20</td>
<td>10</td>
</tr>
<tr>
<td>AUROC</td>
<td>0.862796</td>
<td>0.862251</td>
<td>0.863148</td>
</tr>
<tr>
<td>STD</td>
<td>0.000959</td>
<td>0.000712</td>
<td>0.000502</td>
</tr>
</tbody>
</table>

Table 4.4: Results of the epoch and batch-size grid search, presenting the best results picked by the grid search of epochs and batch-size, with the batch-size being the reference.
4.2 Final Model

Through the grid search, the final model was chosen:

<table>
<thead>
<tr>
<th>Batch-size</th>
<th>Epochs</th>
<th>Layers</th>
<th>Optimizer</th>
<th>Learning Rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>256</td>
<td>10</td>
<td>2 (28,28)</td>
<td>Adam</td>
<td>0.0001</td>
</tr>
</tbody>
</table>

Table 4.5: The configuration of the final model.

Figure 4.1: The ROC curve for the final model, with its AUROC number displayed.

Comments: The plot is auto generated using the scikit-learn library. Looking at the ROC curve of class 1 (class 1 implying that the person is deceased), it was able to have a true positive rate of 0.6 while maintaining a false positive rate of about 0.1. This combined with the AUROC shows that the predictive power of the network is not random.

4.2.1 Performance

As mentioned in the Background, the equation we are trying to model is:
This is the equation for one year mortality rates of a population group given their age, at a certain year. The multi-layer perceptron was tasked to output the probability of any individual dying during the year 2018, by filtering the data on various features of the input data, and averaging the outputs, one can examine mortality rates of different population groups.

4.2.2 Entire Population Group

![Mortality (Entire Test Population)](image)

Figure 4.2: Showing the mean predicted one year mortality rate and actual one year mortality rates based on age.

Comments: Looking at figure 4.2, the predicted one year mortality rates generally increased as age increased, which corresponds well to the aging factor in natural death.
4.2.3 Population Group Divided into Sex

Figure 4.3: Showing the mean one-year predicted mortality rate and the actual one-year predicted mortality rate for each sex.

Comments: Looking at figure 4.3, different sexes have different mortality rates. The predicted one-year mortality rate for females is higher than the predicted one-year mortality rate for males, which reflects on the statistical fact that females live longer than men.
4.2.4 Population Group Divided into Industry

Figure 4.4: Showing the mean predicted one year mortality rates based on age and the four branches of industry which reflects on the different pension type.

Comments: Figure 4.4 suggests that the group of people in (Industry = PA16) and (Industry = SAFLO) in general have a higher one year mortality rate than the group of people in (Industry = KAPKL) and (Industry = ITP). The mortality rates of KAPKL and ITP are very similar, while PA16 has the highest mortality until age = 73 where SAFLO overtakes the PA16 curve.
Industry type: PA 16

Figure 4.5: Graphing the hold-out set, showing the mean predicted mortality based on age and industries which fall into the PA 16 category.

Industry type: SAFLO

Figure 4.6: Graphing the hold-out set, showing the mean predicted mortality based on age and industries which fall into the SAFLO category.
Industry type: KAPKL

Figure 4.7: Graphing the hold-out set, showing the mean predicted mortality based on age and industries which fall into the KAPKL category.

Industry type: ITP

Figure 4.8: Graphing the hold-out set, showing the mean predicted mortality based on age and industries which fall into the ITP category.
4.2.5 Prediction Comparison

![Graph showing predicted mortality rates](image)

Figure 4.9: Graphing the hold-out set, showing the predicted mortality rate curve plotted versus that of Svenska Försäkringar’s (SF) prediction.

In figure 4.9, the networks prediction versus the prediction that Svenska Försäkringar made in 2007 for the population in 2020 is graphed. Svenska Försäkringar only provided data points every five years from age 30 and above. Thus an exponential interpolation was done in Excel to provide data-points to construct this graph. Therefore, for a better comparison, the actual numerical values are presented:

<table>
<thead>
<tr>
<th>Age</th>
<th>Network Prediction, $q_x$</th>
<th>SF Prediction, $q_s$</th>
<th>Difference($q_x - q_s$)</th>
</tr>
</thead>
<tbody>
<tr>
<td>30</td>
<td>8.75E-05</td>
<td>0.000085</td>
<td>2.5E-06</td>
</tr>
<tr>
<td>35</td>
<td>0.000145</td>
<td>0.00018</td>
<td>-0.000035</td>
</tr>
<tr>
<td>40</td>
<td>0.000268</td>
<td>0.000315</td>
<td>-0.000047</td>
</tr>
<tr>
<td>45</td>
<td>0.00052</td>
<td>0.000665</td>
<td>-0.000145</td>
</tr>
<tr>
<td>50</td>
<td>0.000959</td>
<td>0.0012</td>
<td>-0.000241</td>
</tr>
<tr>
<td>55</td>
<td>0.001797</td>
<td>0.002245</td>
<td>-0.000448</td>
</tr>
<tr>
<td>60</td>
<td>0.00327</td>
<td>0.0034</td>
<td>7E-05</td>
</tr>
<tr>
<td>65</td>
<td>0.005858</td>
<td>0.00566</td>
<td>0.000198</td>
</tr>
<tr>
<td>70</td>
<td>0.01052</td>
<td>0.009635</td>
<td>0.000885</td>
</tr>
<tr>
<td>75</td>
<td>0.01787</td>
<td>0.01607</td>
<td>0.0018</td>
</tr>
</tbody>
</table>

Table 4.6: Comparing the network predicted values against Svensk Försäkring’s predicted values.
Chapter 5

Discussion

5.1 Methodology Limitations and Decisions

In general, there were time and resource limitations for the experimentation. The experiment was done on a 4th generation Intel i7 quad-core laptop and there were no available computers with appropriate GPUs for GPU accelerated machine learning which resulted in considerably slower learning times. With regards to the sensitive nature of the data, it was also unfeasible to transfer the data to my own personal working environment with can utilize Nvidia CUDA or any cloud computing environments suited for machine learning. Thus, with regards to the slow learning times, only the multi-layer perceptron was tested even if it became apparent during the literature review that radial basis function network and Bayesian networks could also be appropriate.

According to James et al. [21], it is a standard to use 5 or 10 folds in cross validation due to empirical evidence that models generated from 5 or 10 folds yield results that have neither high variance or bias. However, higher amount of folds imply longer training times and again due to the time constraint, a 3 fold was chosen.

For the grid search ordering, it was done logically through the process of first drawing an appropriate gradient descent surface (layers and nodes), finding which algorithm with its respective learning rate parameter to use for the descent of such a surface (optimizer and learning rate) and finally, how often this surface should update and how many times this descent should be repeated to achieve satisfactory results (batch size and epochs).
5.2 Network Performance and Meta-Analysis

Figure 4.1 showed that the network had an AUROC of 0.81. Hosmer and Lemeshow [23] suggests this result is within the range of "excellent", with AUROC scores of 0.70 to 0.80 being 'acceptable', 0.80 to 0.90 being 'excellent' and 0.9 to 1 being 'outstanding'. Tape [24] also introduces a rule of thumb rating AUROC scores, where this score would receive a B score if one were to use the traditional academic point system. Thus the network tested in this study is indeed deemed highly appropriate to use for modelling one-year mortality rates.

Looking at figure 4.2 it appears that the model was able to successfully generalize over the population as a whole as it was able to learning the age dependant factor. Figure 4.3 demonstrated the models ability to learn the sex dependant factor, in which both the predicted and actual mortality rates showed that females on average live longer. This is backed by further evidence by the original Makeham formula, which included sex factor that translated the difference between male and female mortality and deducted it from the time(age) factor. Both age and sex were also factors included in the networks deployed by Shi, Wang, and Wang [6] and Monsalve-Torra et al. [7]. Figure 4.4 suggests that at the age of 75, there is a noticeable gap between the PA16, SAFLO, KAPKL and ITP mortality rates. With PA16 and SAFLO being higher, while KAPKL and ITP being lower. This is interesting because PA16 is a pension represented by government employed workers such as: police, military, university teacher, air traffic controller etc. SAFLO pension is represented by private workers such as: construction workers, hotel employee, restaurant employee, factory worker etc. In general, one can argue that many of these employment types are typically characterized labor or mentally intensive which contributes to the general deterioration of health. A statistical analysis from a study[25] as recent as 2018 shows that a high level of physical activity in ones occupation had an 18% increased risk of early mortality.

Table 4.6 showed the network predicted values against the values predicted by Svensk Försäkring. Looking at the differences between $q_s$ and $q_s f$, it demonstrates that the predictive power of the network is comparable to that of the predictions done by Svensk Försäkring, with very small differences through all ages. This table and graph 4.9 shows that the network had lower predicted one year mortality rates between ages 35 to 60, while from age 60 and above, the network showed higher mortality rates than the predictions made by Svensk Försäkring. Looking at the actual mortality rates from figure 4.8 given by the test set, it is inconclusive which one performed better.
However the spikes from figure 4.8 may indicate that the sample size of ITP categorized individuals were low, and non-representative of how actual one year mortality rates should actually look given a large enough sample size.

### 5.3 Future Studies

In the literature study, Shi, Wang, and Wang [6] and Monsalve-Torra et al. [7] showed that medical conditions are feature vectors that would have an impact on the one year mortality rates. The implications of adding medical conditions, such as smoking [26] could further discriminate individuals into smaller groups that would represent their one-year mortality rates to a more accurate degree. Folksam, a Swedish insurance company, have an semi-automated process where one can acquire a whole life insurance. In this process, they ask about the medical history about the applicant[27]. Examples are whether the individual uses prescription based medicine, smoking status, height, weight, and any of 14 mentioned medical conditions for which the applicant has had to apply professional care for. This is further evidence that health factors impact the mortality rates. Thus, if possible, incorporating health factors into a one year mortality prediction neural network for insurance purposes should be largely beneficial into narrowing down a large population group into a lot more narrow groups with different risks.

Due to time constraints, this report did not evaluate radial basis function networks and Bayesian networks which were shown to be a possibility in the literature review. Therefore, a suggestion for future work is to investigate the application of the aforementioned networks to one year mortality probability rates and compare their performance to multilayer perceptrons.

A further extension of this study could be done to predict how one year mortality rates would look in ten years. This would involve having a large and extensive data set that would date back many decades, and a time series analysis could be done to study how the one year mortality rates would differ from year to it’s previous years.
Chapter 6

Conclusions

The network itself yielded an AUROC value that is considered "excellent" with a score of "B" according to two generally accepted ratings of AUROC values. The network was able to learn that mortality increases with age, as well as that women on average live longer than men and that there was a difference in the one year mortality rates regarding the different industries that constituted the four different types of occupational pensions that was examined in this study. Comparing to the predictions made by Svensk Försäkring, it performed on a similar level but conclusions could not be drawn as to which one performed better.

Suggestions for future work based on this report were made. The literature review suggests that adding health factors could further improve the prediction of one year mortality rates, adding the ability to divide the population into even smaller groups. Using Radial basis function networks and Bayesian networks to tackle the same task should also be evaluated as these networks might offer better performance compared to multilayer perceptrons.
Bibliography


Appendix A

Data dump of the grid search results conducted in the order explained in the methodology.

Best: 0.863001 using {'neurons_first_layer': 28}
0.738764 (0.168834) with: {'neurons_first_layer': 3}
0.861710 (0.002258) with: {'neurons_first_layer': 7}
0.862885 (0.002149) with: {'neurons_first_layer': 14}
0.863001 (0.002184) with: {'neurons_first_layer': 28}

Best: 0.863949 using {'neurons_first_layer': 28, 'neurons_second_layer': 28}
0.860942 (0.001206) with: {'neurons_first_layer': 3, 'neurons_second_layer': 3}
0.860533 (0.003564) with: {'neurons_first_layer': 3, 'neurons_second_layer': 7}
0.862282 (0.001441) with: {'neurons_first_layer': 3, 'neurons_second_layer': 14}
0.862885 (0.002149) with: {'neurons_first_layer': 3, 'neurons_second_layer': 28}

Best: 0.862926 using {'neurons_first_layer': 28, 'neurons_second_layer': 3, 'neurons_third_layer': 28}
0.741370 (0.170675) with: {'neurons_first_layer': 3, 'neurons_second_layer': 14, 'neurons_third_layer': 3}
0.861164 (0.000738) with: {'neurons_first_layer': 3, 'neurons_second_layer': 14, 'neurons_third_layer': 7}
0.860226 (0.002202) with: {'neurons_first_layer': 3, 'neurons_second_layer': 14, 'neurons_third_layer': 14}
0.861795 (0.001403) with: {'neurons_first_layer': 3, 'neurons_second_layer': 14, 'neurons_third_layer': 28
Best: 0.863148 using {'batch_size': 256, 'epochs': 10}
0.861027 (0.000994) with: {'batch_size': 64, 'epochs': 5}
0.862796 (0.000959) with: {'batch_size': 64, 'epochs': 10}
0.862672 (0.000322) with: {'batch_size': 64, 'epochs': 20}
0.862339 (0.001130) with: {'batch_size': 64, 'epochs': 40}
0.861624 (0.001066) with: {'batch_size': 128, 'epochs': 5}
0.862251 (0.000464) with: {'batch_size': 128, 'epochs': 10}
0.863128 (0.000712) with: {'batch_size': 128, 'epochs': 20}
0.862931 (0.000715) with: {'batch_size': 128, 'epochs': 40}
0.862994 (0.000807) with: {'batch_size': 256, 'epochs': 5}
0.863148 (0.000502) with: {'batch_size': 256, 'epochs': 10}
0.862789 (0.000732) with: {'batch_size': 256, 'epochs': 20}
0.862807 (0.000221) with: {'batch_size': 256, 'epochs': 40}