UNIVERSITÀ CATTOLICA DEL SACRO CUORE

MILANO

Faculty of Banking, Finance and Insurance Sciences

Master of Science in Statistical and Actuarial Sciences

PROJECTION OF MORTALITY TABLES: A NEURAL NETWORK APPROACH

Candidate: PIROVANO Simone

Matr. 4802914

Supervisor: Prof. CLEMENTE Gian Paolo

Academic Year 2019-2020

Contents

Abstract

Some aspects of the actual society are planned according the values of expected future mortality rates. In life insurance industry, pension plans, and social security schemes have deal with future cash flows, related to survival and future longevity dynamics.

In the past two centuries, in the world, the life expectancy has more than doubled. For the importance of the argument and for the mutability of this variable, many models that estimate and forecast the future mortality rates have been proposed. Among them, one of the most influential is the Lee-Carter model.

The aim of this master thesis is to forecast future mortality rates. The approach that will be used is based on the classical Lee-Carter model. The traditional model that is based on the projection of the time parameter using an ARIMA model suffers a lot of limitations.

In this thesis it has been presented a Neural Network extension of the Lee-Carter model, using the innovative Recurrent Neural Network called Long-Short Term Memory (LSTM) and Gated Recurrent Unit (GRU) architectures. The approach has shown to improve the predictive ability and be much more flexible.

A detailed comparison of the three models have been carried on different countries in order to understand if this new approach could be useful and generalizable to different countries all over the world.

Chapter 1

Introduction

"We may regard the present state of the universe as the effect of its past and the cause of its future. An intellect which at a certain moment would know all forces that set nature in motion, and all positions of all items of which nature is composed, if this intellect were also vast enough to submit these data to analysis, it would embrace in a single formula the movements of the greatest bodies of the universe and those of the tiniest atom; for such an intellect nothing would be uncertain and the future just like the past would be present before its eyes. "

Pierre Simon Laplace (1825) – A Philosophical Essay on Probabilities

Life expectancy has changed during the time as response to the technological, medical and social changes. As an example, in 1870 life expectancy at birth in Italy was only about 30 years and then it has increased till 82 years in 2020. The improvement has not been constant during the time, in fact, due to the medical and biological development of the last 150 years we have experienced a higher development. These changes in mortality and longevity are not uniform with respect to the ages: the rectangularization and expansion movement of the survival curve seems to explain it very well. Until now, the improvement in longevity seems to not have reasons to slow down, thanks also to the new discoveries in medicine.

Some aspects of the modern society are planned according the values of expected future mortality rates. Life insurance industry, pension plans, and social security schemes have to deal with contracts that links benefits to the duration of human life. Longevity risk is

the risk that people lives longer than expected and correspond to one of the major risks for this kind of companies. In this context, Actuaries and Demographers have developed a lot of methodologies to predict the future probabilities regarding event of death, life and linked quantities.

One of the most relevant models is the Lee-Carter model, developed in 1992. Several extensions of this method have been proposed during the last years. In this master thesis, the classical Lee-Carter model will be extended using the Recurrent Neural Networks models, in particular, Long-Short Term Memory and Gated Recurrent Unit.

In this way it will be provided a quantitative comparison of the Classical Lee-Carter that utilizes the ARIMA model to project the time dependent variable and the models based on the Neural Networks architecture. The comparison is based on different aspects: an error measurement on the forecast of the Lee-Carter's time dependent variable, the forecast on the probability of death, the curve of deaths and the calculation of different annuities.

Chapter 2

Introduction to mortality models

The aim of this chapter is to create solid basis to understand how mortality is evolving and which are the techniques used to project mortality in the time. In demography, mortality is studied using life tables which will be described in chapter 2.1 differencing from cohort life table and period Life table (2.1.1).

Next, it will be explained the main notations regarding mortality in chapter 2.2.1 and the life probabilistic framework in chapter 2.2.2 and some interesting market indicator in chapter 2.2.5.

In chapter 2.3 there are described some famous and recurrently used mortality laws.

Finally, in chapter 2.4 it's described how to move from a basic model to a more general model introducing and describing some risks factors.

2.1 Life tables

Life tables are used to represent the mortality (survival) of the population. In particular, they are represented by a set of decreasing finite sequence of numbers denoted by l_0, l_1, l_2, …, l_ω. The element l_x represents the estimated (rounded) numbers of people alive at age x in a properly defined population.

The element l_0 is called radix of the tables and normally it's 100000, it represents the initial number of individuals.

The ages are defined as x = 0, 1, 2, …, ω, where ω is the maximum attainable age, the upper limit, i.e. where $l_\omega>0$ and $l_{\omega+1}=0$.

Starting from l_x it can be defined the number of deaths for the age x denoted as d_x for x=0,1,2, …, ω.

$$d_x=l_x - l_{x+1} \textit{ (formula 2.1)}$$

notice that:

$$\sum_{x=0}^{\omega} d_x = l_0 \textit{ (formula 2.2)}$$

It can be observed that the l_x is a monotonic decreasing function, while the d_x is not strictly monotonic.

The table 2.1 represents a glimpse of a Life table as it has been described previously.

x	l_x	d_x
0	100 000	879
1	99 121	46
2	99 076	33
…	…	…
50	93 016	426
51	92 590	459
…	…	…
108	1	1
109	0	0

Table 2.1: Life Table example (SOURCE: ISTAT)

In the end, it should be mentioned that there are two kind of life tables: Cohort Life Tables and Period Tables (Chapter 2.1.1).

2.1.1: Cohort and Period Life Tables

Cohort life table is a kind of life table in which we directly observe, as statistical evidence, the sequence l_0, l_1, l_2, …, l_ω, starting from an initial cohort consisting of l_0 newborns: this

constitutes a cohort of people born in the observed year t. Then from this cohort we can observe longitudinally year by year, creating different cohort for each observed year.

Period life table are built assuming to observe directly the frequency of death at various age, which represents the probability to die for a given period, for example a year.

Once the values of age-specific mortalities are estimated for each x=0,1,2, …, ω, possibly with a smoothing effect with respect to x and denoted as q_x.

Then for x=0,1,2, …, ω, we define:

$$l_{x+1} = l_x (1 - q_x) \quad \textit{(formula 2.3)}$$

After having assigned l_0, which is the radix (usually as l_0=100000) then the sequence l_0, l_1, l_2, …, l_ω is computed using the previous recursive formula (formula 2.3).

This kind of life table is called Period Life Table and it's derived from period mortality observations.

The most important point to be highlighted is that the frequency of deaths are observed in an year which is different from the year in which we are going to estimate the l_x This means that this methodology is hypotising that the mortality pattern isn't changing during the years.

As it has been expressed in the Introduction and as the statistical evidence shows, the mortality in many countries is declined rapidly and the hypothesis of "static" mortality on witch this methodology rely on cannot be assumed as a principle in the long run.

For this reason, this kind of life tables cannot be used for long duration life product, such as annuities, while can be used for short term contracts.

In the following plot there is an example of survival curve (figure 2.1) deaths curve (figure 2.2).

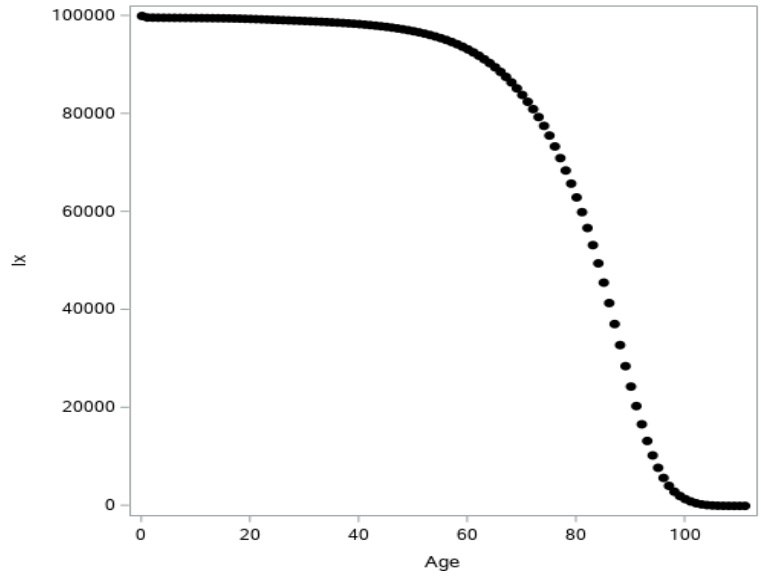

Figure 2.1: Survivals Curve l_x for male population in 2018 – SOURCE: ISTAT

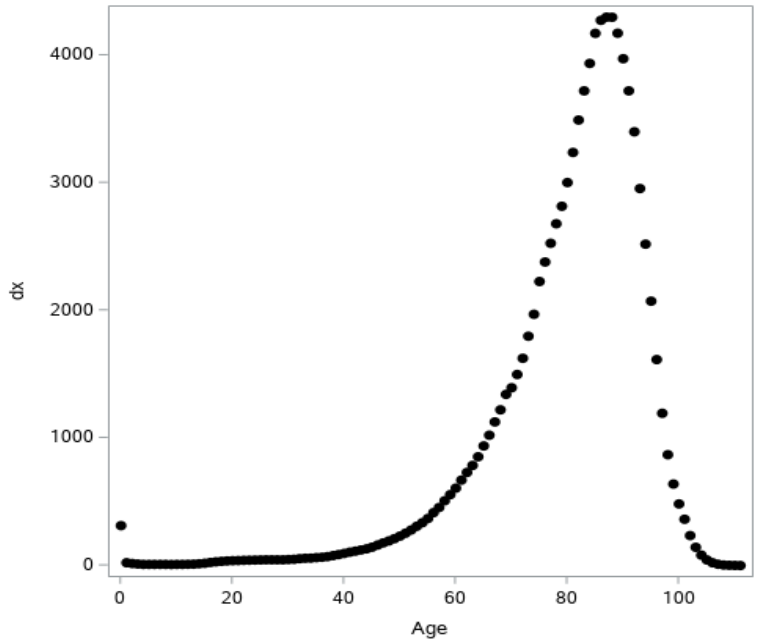

Figure 2.2: Deaths curve d_x for male population in 2018 – SOURCE: ISTAT

The Survival Curve (fig.2.1) represents on the y axis the value of l_x (number of survival death for the age x) and the age x on x axis.

The Deaths Curve (fig.2.2) represents on the y axis the value of d_x (number of deaths for the age x) and value of the age x on the x axis.

There are some features that can be observed in the deaths curve:

1. The infant mortality;

2. The mortality hump at young-adult ages, mainly due to accidental deaths;
3. The age of maximum mortality (at old ages);

It can be noted in addition that the point with the highest mortality is where the survival curve inflex and it's around 89 year for a male in Italy in 2018.

2.2 The probabilistic Model

In this chapter it's formalized the probabilistic model underlying the random lifetime of an individual. It's useful to model the mortality risk and build more sophisticated projection considering also an interval of confidence, in addition to the simple expectation.

Finally, in the following paragraphs are presented some interesting and useful markets indicators and formulas regarding mortality.

2.2.1 Basic Notations and Definitions

In this chapter it will be provided the basic notation and definition of life and death probabilities that will be used then in the next chapters.

The simplest probability, used in this context, is the annual probability of survive p_x which is defined as follows:

$$p_x = \frac{l_{x+1}}{l_x} \quad \textit{(formula 2.4)}$$

The annual probability to die can be defined as:

$$q_x = 1 - p_x = \frac{d_x}{l_x} = \frac{(l_x - l_{x+1})}{l_x} \quad \textit{(formula 2. 5)}$$

The survival probability for an individual with a generic age x for t years, in other words the probability for an individual of x years to reach the age x+t, is defined as ${}_tp_x$ and by following the definition provided by the life tables can be calculated as:

$${}_tp_x = \frac{l_{x+t}}{l_x} \quad \textit{(formula 2.6)}$$

The formula 2.6 can be seen also as product of annual probabilities, we can define it as follows:

$$ {}_tp_x = \prod_{s=1}^{n} p_{x+s} \quad \textit{(formula 2.7)} $$

Another probability related to death event is the probability of death for an individual of a generic age x and for t years is defined as:

$$ {}_tq_x = 1 - {}_tp_x = \frac{d_{x+t}}{l_x} = \frac{(l_x - l_{x+t})}{l_x} \quad \textit{(formula 2.8)} $$

In addition to these basic probabilities we can find also the probability to die between t and t+m, which is ${}_{t/m}q_x$, calculated as:

$$ {}_{t/m}q_x = {}_tp_x \cdot {}_mq_{x+t} \quad \textit{(formula 2.9)} $$

2.2.2 The probabilistic framework: random lifetime

A more formal approach concerning life probabilities will be presented in this chapter, this method is then useful for risk analysis since here it is considered life as a random variable and for this reason we can find not only a simple expectation, but also a distribution of probabilities for the event regarding life or death that we are going to consider.

This approach takes into consideration the life tables calculated as Period Life Tables (chapter 2.1.1) and it considers time as continuous, so age (x) and time (t) will not just be considered as integer but they can take any number (i.e. continuous context).

Starting from T_0 which is the remaining lifetime for an individual aged 0, in other words, a newborn it can be then defined T_x which is the random variable that represent the remaining lifetime of an individual aged x.

$$ T_x = T_0 - x \mid T_0 > x \quad \textit{(formula 2.10)} $$

From these definitions it can be defined the Survival Function which is the probability of survive t years for an individual aged 0 years as follows:

$$ S_0(x) = {}_tp_0 = \text{Prob}(T_0 > t) = \frac{l_{x+t}}{l_x} \quad \textit{(formula 2.11)} $$

Another interesting function is the Distribution Function of T_0 which is defined as:

$$ F_0(x) = {}_tq_0 = \text{Prob}(T_0 \leq t) = \int_0^x f_0(x)dx \quad \textit{(formula 2.12)} $$

This function represents the probability of die within x years for a newborn, under the probabilistic framework.

The probability density function of T_0 is $f_0(t)$ and it's equal to the derivatives of $F_0(x)$; this function is often defined "curve of deaths".

$$f_0(t) = \frac{dF_0(x)}{dx} = -\frac{dS_0(x)}{dx} \text{ (formula 2.13)}$$

Then moving to more complete function we can find the probability to survive t years for an individual aged x:

$${}_tp_x = Prob(T_x > t) = Prob(T_0 > x + t | T_0 > x) = \frac{Prob(T_0 > x+t)}{Prob(T_0 > x)} = \frac{S_0(x+t)}{S_0(x)} \text{ (formula 2.14)}$$

The probability of die within t years for an individual aged x instead is:

$${}_tq_x = Prob(T_x \leq t) = 1 - Prob(T_0 > x + t | T_0 > x) = 1 - \frac{Prob(T_0 > x+t)}{Prob(T_0 > x)} = \frac{S_0(x) - S_0(x+t)}{S_0(x)}$$

(formula 2.15)

These probabilities have been defined using a probabilistic framework, the plot of the curve of death and of the survival function is in the next figures and as it can be seen both functions are continuous due to the hypothesis we have made.

Here in after are plotted an example of survival curve (figure 2.1) death curve (figure 2.2)

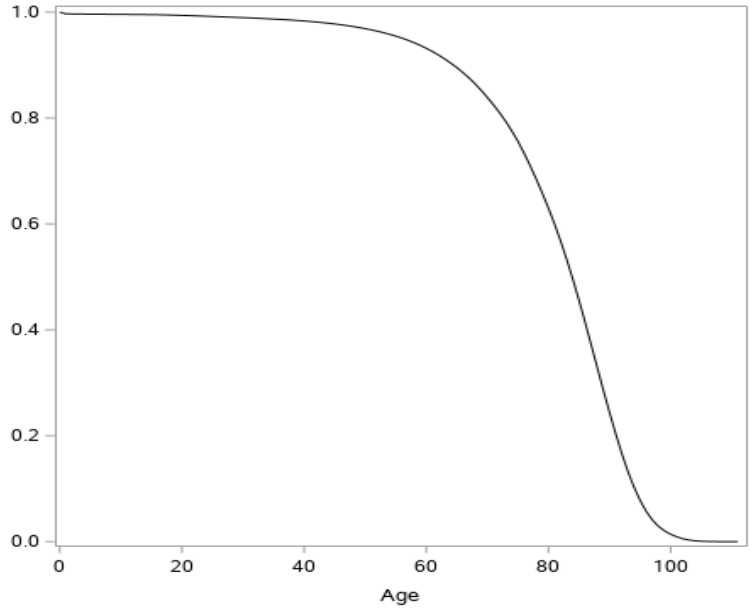

Figure 2.3: Survival Curve $S_0(x)$ for Italian male population in 2018 – SOURCE: ISTAT

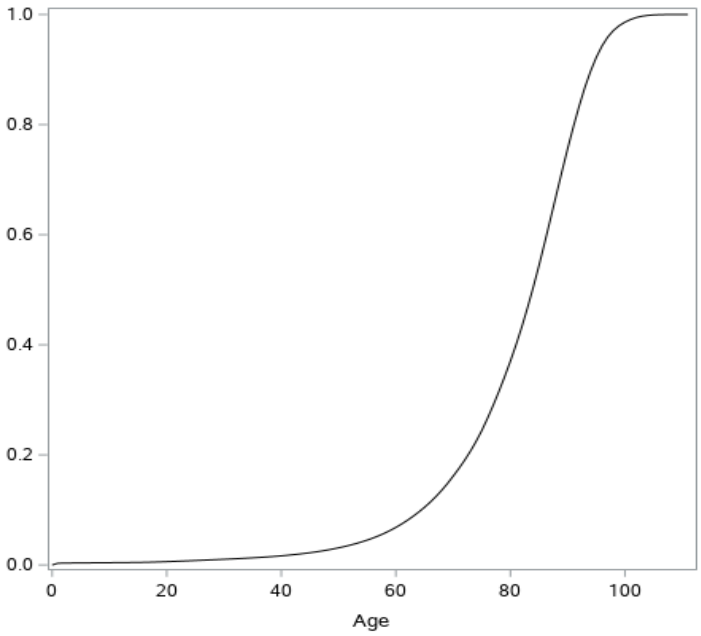

Figure 2.4: Distribution Function of T_0 called $F_0(x)$ for Italian male population in 2018 – SOURCE: ISTAT

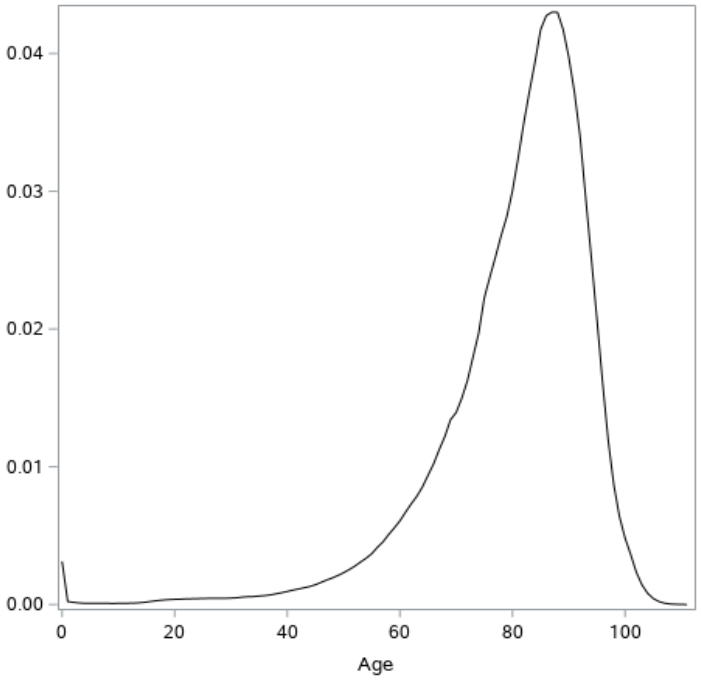

Figure 2.5: Curve of death $f_0(t)$ for Italian male population in 2018 – SOURCE: ISTAT

2.2.3 Force of mortality and central rate of mortality

There are some indicators which are very important and useful in demography, some of them are used also to project the life tables in a future time. In this chapter there will be presented some of these indicators.

The first indicator is the force of mortality μ_x which is the instantaneous rate of mortality of a given age x. The force of mortality can be denoted as function of the survival function, in fact can be written in terms of $S_0(x)$.

$$\mu_x = \lim_{x\to 0} \frac{\text{Prob}(T_x \le t)}{t} = \lim_{x\to 0} \frac{\text{Prob}(T_x \le t)}{t} = \lim_{x\to 0} \frac{F_0(x+t) - F_0(x)}{S_0(x)} = \lim_{x\to 0} \frac{S_0(x) - S_0(x+t)}{S_0(x)} = -\frac{S_0'(x)}{S_0(x)} = \frac{f_0(x)}{S_0(x)}$$

(formula 2.16)

Then it follows that:

$$S_0(x) = \exp\{-\int_0^x \mu_x \, dz\} \quad \textit{(formula 2.17)}$$

Another interesting and very useful indicator is the central rate of mortality. It summarizes the behavior of the force of mortality over a given interval and it's denoted by $m_{(x,x+t)}$. It's defined as:

$$m_{(x,\,x+t)} = \frac{\int_0^t S_0(x+u)\,\mu_{x+u}\,du}{\int_0^t S_0(x+u)\,du} \quad \textit{(formula 2.18)}$$

In the most used case of the annual central rate of mortality we have:

$$m_x = \frac{\int_0^1 S_0(x+u)\,\mu_{x+u}\,du}{\int_0^1 S_0(x+u)\,du} \quad \textit{(formula 2.19)}$$

this indicator summarizes the behavior of the force of mortality in a one-year framework.

By using the trapezoidal approximation for the integral at denominator:

$$\int_0^1 S_0(x+u)\,du = \frac{S_0(x) + S_0(x+1)}{2} \quad \textit{(formula 2.20)}$$

It can be proven that:

$$m_x \cong \frac{S_0(x) - S_0(x+1)}{\frac{S_0(x) + S_0(x+1)}{2}} = \frac{2q_x}{2 - q_x} \quad \textit{(formula 2.21)}$$

and the inverse formula is:

$$q_x = \frac{2m_x}{2 + m_x} \quad \textit{(formula 2.22)}$$

In the following figures there are the plot of the m_x and the μ_x respectively.

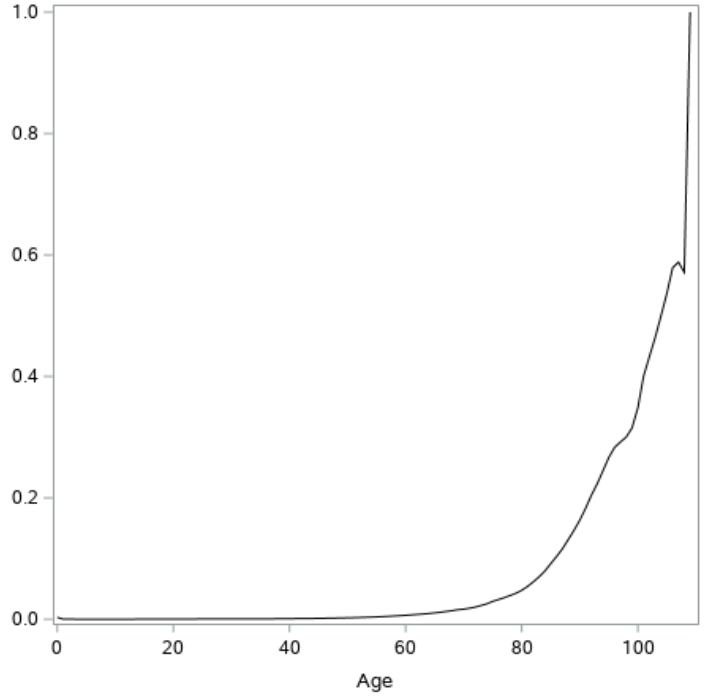

Figure 2.6 Force of mortality μ_x for Italian male population in 2018 – SOURCE: ISTAT

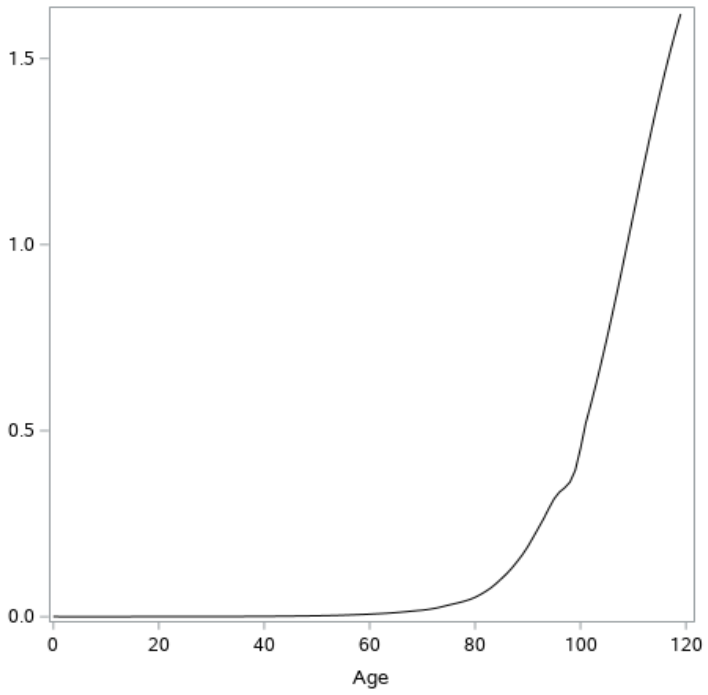

Figure 2.7: Annual central rate of mortality mx for Italian male population in 2018 – SOURCE: ISTAT

As can be easily noticed the force of mortality for the Italian male population in the 2018 is at the beginning till 50 years mostly flat, then clearly increase in the time, we can observe a little decrease at extreme age (i.e. bigger than 100 years). This phenomenon can be caused by the fact that few people reach ages bigger than 100 and so we have few data and a lot of volatility.

As opposite, the annual central rate of mortality is mainly monotonically increasing.

2.2.4 The basic model

Given the probabilistic framework that it has been assumed in the last section, it's useful to try to estimate the simplest model regarding number of survivals.

In facts it's possible to estimate expected number of survivals given an initial number L_a, this model is considering only the age factor, indeed, exists different risk factors, this model considers only standard risks.

It's possible to assume that:

L_a is the number of people alive at age a.

$T_a^{(j)}$ is the random variable regarding the remaining lifetime of an individual j aged a.

$$I_X^{(J)} = \begin{cases} 1 \; if \; T_a^{(j)} > x - a \\ 0 \; if \; T_a^{(j)} \leq x - a \end{cases} \quad \textit{(formula 2.23)}$$

$I_X^{(J)}$ is the indicator function regarding the survival condition of the j^{th} person alive at age a, it's assumed to be Bernoulli distributed.

It follows that:

$$\mathrm{E}(I_X^{(J)}) = 1 \cdot Prob\left(T_a^{(j)} > x - a\right) + 0 \cdot Prob\left(T_a^{(j)} \leq x - a\right) = {}_{x-a}p_a \; \textit{(formula 2.24)}$$

Then it can be found that:

$$I_X^{(J)} \sim Ber\,({}_{x-a}p_a)$$

The survival status of an individual aged a till age x is distributed as a Bernoulli with the probability ${}_{x-a}p_a$.

Hence given that L_x is the random number of people alive at age x.

By assuming that $I_x^{(J)}$ are independent and identically distributed random variables for each j^{th} element, it can be calculated the expected value of survival aged a today with x age in the future.

$$E(L_x) = E(\sum_{J=1}^{L_a} I_x^{(j)}) = \sum_{J=1}^{L_a} E(I_x^{(j)}) = \sum_{J=1}^{L_a} \quad {}_{x-a}p_a = L_a \cdot {}_{x-a}p_a \textit{ (formula 2.25)}$$

It has been found that the expected future number of people $E(L_x)$ starting from a population of L_a people is equal to the initial population multiplied by the probability of survive of the same population aged a for x-a years.

In the end, it can be found that:

$$L_x \sim \text{Binomial } (L_{a,\ {}_{x-a}p_a})$$

This general model is not considering differences between people and increment or decrement of mortality related to risks factor, indeed this method is taking into account only the attained age of the person to assign the probability of an individual dying within one year.

2.2.5 Market indicators

In demography there are some indicators which can help to understand and summarize the life tables and the probability distribution.

In the following there is a list of them:

1. Life expectancy

This is the expectation of the random remaining life for an individual aged x, it's defined as the following:

$$e_x = \text{EXPECTED VALUE}(T_x) = \sum_{h=0}^{\omega - x} h \cdot {}_{h\backslash 1}q_x \textit{ (formula 2.26)}$$

The expectancy of an individual aged x can be easily quantified as $x + e_x$.

It's important to denote that e_0 is the expected value of total lifetime at birth and this level is 82 years for a newborn in 2018 in Italy. This value is increased a lot in the time, in fact in 1975 was only about 69 years.

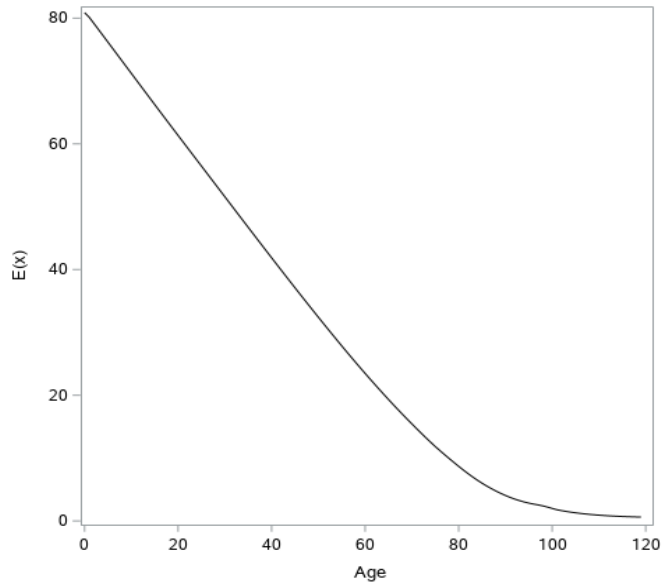

Figure. 2.8 Expected remaining lifetime for an individual aged x for Italian male population in 2018 – SOURCE: ISTAT

2. Lexis point

This indicator represents the modal value of the probability distribution of deaths at old age, in other words, it's the point where there is the higher number of deaths.

3. Variance of the probability distribution

This indicator is very important to understand the volatility of the distribution of the probability to die, survive.

4. Coefficient of variation of the probability distribution

This indicator represents a measure of volatility of the distribution of the probability to die, it's a measure of relative volatility and it's very used in practice.

2.3 Mortality laws

Actuaries and demographers have been interested into find an analytical formula to describe the mortality, in other words a method that fits well the age pattern of mortality without using data that comes directly from the empirical observations.

This formula is called mortality law and it's useful to describe the longevity by means of a small number of parameters, reducing the dimensionality of the problem and simplifying the calculations.

There are many ways to specify a mortality law, some models find a law for the l_x, some others use the force of mortality μ_x. The models will be presented in chronological order.

1725 De Moivre's law:

The model specified from the mathematician Abraham De Moivre.

The model supposes that an extreme age ω exists and it's the only parameter of the model. Moreover, it supposes that the age is contained in an initial age a and .

The survival function $S_0(x)$ is supposed to be linear in the age x, so that:

$$S_0(x) = S_0(a)\frac{\omega - x}{\omega - a} \text{ for } a < x < \omega \quad \textit{(formula 2.27)}$$

$$S_0(x) = 0 \text{ for } x > \omega$$

Then for what concern force of mortality can be found:

$$\mu_x = -\frac{S'_0(x)}{S'_0(x)} = \frac{1}{(\omega - x)} \quad \textit{(formula 2.28)}$$

Gompertz's Law 1824 and Makeham's Model 1860

Both the models base on an exponential mortality law, in fact they suppose that the force of mortality increases exponentially:

$$\mu_x = \beta c^x \quad \textit{(formula 2.29)}$$

This is the case of the Gompertz's Law, then Makeham has added an additional constant term to this model:

$$\mu_x = \beta c^x + \alpha \quad \textit{(formula 2.30)}$$

This additional term is used to account accidental death independent from the ages.

Starting from:

$$S_0(x) = \exp\{ - \int_0^x (\alpha + \beta c^s) ds \} \quad \textit{(formula 2.31)}$$

Integrating and saying that s= $e^{-\alpha}$, g=$e^{-\frac{\beta}{log(c)}}$, 0<g<1, we can obtain the survival function as:

$$S_0(x)=s^x g^{c^x} \text{ (formula 2.32)}$$

To this the probability for a person aged x to survive t years more is:

$$_tp_x=\frac{S_0(x+t)}{S_0(x)}=s^x g^{c^x(c^t-1)} \text{ (formula 2.33)}$$

In general, Gompertz and Makeham mortality laws are good model for big intervals of ages such as 25/30 to 95. Anyway, both models are not able to capture the reduction of infant mortality neither the increase of mortality due to accidents, in ages between 18 and 25, this phenomenon is due to the monotonic increasing behavior of the force of mortality.

1878 Dermoy's Model:

This model supposes the force of mortality constant in the time as $\mu_x = \mu$ in this way we can obtain an exponential survival function as:

$$S_0(x)= S_0(x)e^{-\mu(x-a)}$$

$$S_0(x)=e^{-\mu(x-a)} \text{ for } S_0(a)=1 \text{ and } a=0 \text{ (formula 2.34)}$$

The main issue of this function is that approximates the survival function only for very short interval of ages.

1980 The Heligman- Pollard Law:

This law is important in the dynamic context since often is useful to describe the observed mortality trend and predict the age pattern of mortality in the future.

We define the ratio $\frac{q_x}{p_x}$ as the measure for the maximum improvement possible of the survival function for an individual aged x with respect to the value of the function in that age x.

The model is specified as:

$$\frac{q_x}{p_x}=A^{(x+B)^c}+D^{-E(\log(x)-\log(F))^2}+G\cdot H^x \text{ (formula 2.35)}$$

This formula is composed by three blocks which are added together and represents three moments of the life.

The first addend represents the behavior of mortality in infant ages.

- A is approximatively equal to q_1
- B depends by the ratio between $q_1 \, and \, q_0$
- C expresses the speed of decreasing of the infant mortality

The second addend represents the accidental mortality of the medium ages.

- U is the scale parameter relating to the peak of accidents
- E is the parameter regarding the distribution of the peak of accidents
- F is the parameter regarding the position of the peak

The third addend describe the mortality for the old ages.

- G express the mortality at the beginning of aging process
- H is the measure of the speed of increasing of probability to die getting bigger the ages

Through this formula we can describe a lot of phenomena using only eight parameters.

2.4 From basic model to more general models

In the chapter 2.2 it has been explained a basic model regarding mortality, that method was considering only the age effect of the person, anyway there are also additional effects that influence the death probability.

Looking at the data, for instance, it can be noticed that people who smokes are more likely to get lung cancer and maybe die before that a person who doesn't smoke.

For this reasons, in many situation, the basic model is not able to explain the reality but actuaries and demographers have created models allowing for heterogeneity between people, incorporate the future mortality trend, the effects of medical ascertainment in the underwriting process and so on.

In the following chapter it will be analyzed all the different model that incorporate these effects.

2.4.1 Heterogeneity and rating classes

Since inside any population there are differences from a person to another, it's not statistically possible to treat people as they would be identic, this is called heterogeneity

There are two kind of heterogeneity factor, some are observable, others are unobservable.

Observables risk factor are such that can be observed and classified. In this category can be found:

1. Classical biological and physiological factors, such as age, gender, and genotype;

2. Elements of the living environment; in particular: climate and pollution, nutritional

standards (mainly with reference to excesses and deficiencies in diet), population

density, hygienic, and sanitary conditions;

3. Occupation and, more in general, social status;

4. Individual lifestyle, in particular: nutrition, alcohol and drug consumption, smoking, physical activities, and pastimes;

5. Current health conditions, personal and/or family medical history, civil status, and

so on.

In contrast, unobservable heterogeneity factor are not common to everybody but are person specific, for this reason cannot be used to define risk classes. An example of this kind of factor is the individual attitude toward health.

Observable risk factors can be used to define the rating classes which can be useful in pricing, in fact, we can split the population into different rating classes when pricing and define a price for the specific group.

However, not all the risk classes can become risk factors since regulation prohibit to use gender when pricing. This rule under the actuarial point of view is wrong since as can notice by the plot, female person is more likely to survive longer than men.

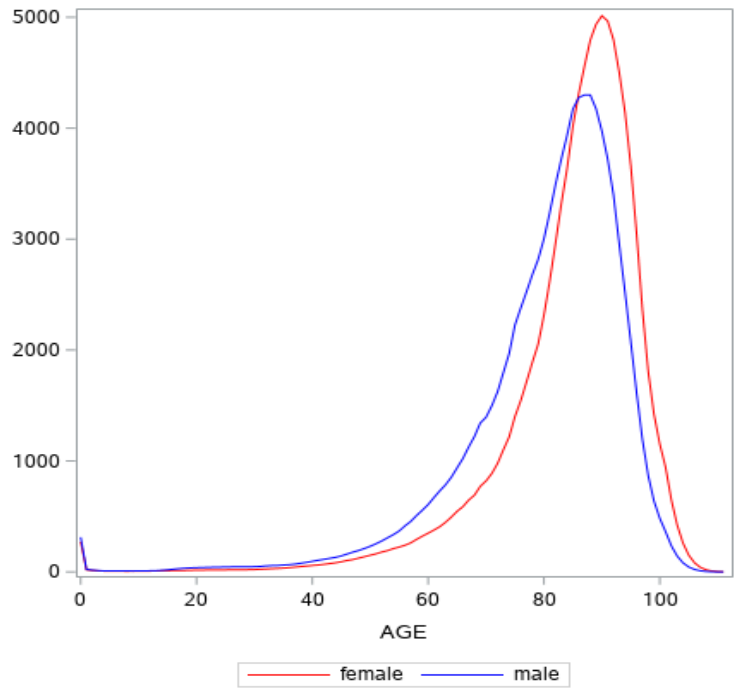

Figure 2.9: Curves of death in the Italian male (blue) and female (orange) populations in 2018– SOURCE: ISTAT

Another interesting effect is called anti-selection or adverse selection.

It means that people which are particularly exposed to a certain risk try to insure that risk. An example of this condition are people which are in bad life status and try to assure their death with term insurance policies; as contrary, people which are in particularly good condition try to insure their life with life annuities.

This phenomenon is avoided and solved by insurer asking to people who wants to insure their life a certificate regarding life status of the person who want to insure himself.

Both these items should be considered when pricing an insurance. The pricing procedure consists firstly in creating the rating factors, then it's necessary to choose the appropriate risk group for the individual, and finally try to solve the problem of adverse selection.

Accounting the different kind of risks, it's possible to make the table more specific. In this way it's possible to find standard and sub-standard risks.

2.4.2 Sub-standard Risks

There are different kind of aggravation methods for the substandard risks, in fact, it can be seen the probabilities of the substandard risks as a transformation of the mortality tables for standard risks, in general aggravated particularly.

In the following paragraph, the age at policy issue is denoted as x, and by m the policy term. In addition, it can be denoted the annual probability of die as q_{x+t} ($0 \leq t \leq m-1$) according to mortality tables used for the standard risk.

A general formulation is the Linear model that is represented as:

$$q^{L}_{x+t} = (1+\gamma)q_{x+t} + \delta q_x \quad \textit{(formula 2.36)}$$

Going into the specific, the Multiplicative form could be specified as:

$$q^{M}_{x+t} = (1+\gamma)q_{x+t}\,, \gamma > 0 \quad \textit{(formula 2.37)}$$

In this way the γ is the percentual aggravation costant and that the aggravation increases in increasing of the mortality and so the age, this model is useful to express biological risks related to ages.

Another interesting form of the more general linear model is the Additive Model, as follows:

$$q^{A}_{x+t} = q_{x+t} + \delta q_x \quad \textit{(formula 2.38)}$$

In this way mortality results constantly aggravated, independently by di age, this can be used to consider occupational risks.

The decreasing extra mortality model is a model very used in practice to represent substandard risks. This representation is used to figure a situation which could be manifested within a specific interval of time and could be terminated with the death or the conclusion of this situation, the end represents the end of the aggravation.

The following relation represents what said above:

$$q^{M}_{x+t} = \begin{cases} (1+\gamma)q_{x+t} + \delta q_x & \text{for } t < r-1 \\ q_{x+t} & \text{for } r \leq t \leq m-1 \end{cases} \quad \textit{(formula 2.39)}$$

Finally, the last model is the age shift model, this model simply associates the probability of an individual aged x to the probabilities of an individual with older ages. This can be represented with the following formula:

$$q^{\text{AGE SHIFT}}_{x+t} = q_{x+t+s} \text{ for } 0 \leq t \leq m\text{-}1 \text{ and } s > 0 \quad \textit{(formula 2.40)}$$

2.4.3 Selected and Ultimate life tables

In the study of the mortality tables some behavior of the individuals have been pointed out and for these reason some modification to the mortality tables can be made.

It's has been observed that the mortality of a person who has recently bought a term insurance (it can be taken as standard mortality) is lower than the mortality of a person who has bought the insurance since a long period.

This fact can be denoted in this way:

$$q_{[x]} < q_{[x-1,1]} < \ldots < q_{[x-n,n]}$$

The first part of the symbols at the subscript is the year in which the policy holder has bought the insurance, while the second part is the number of years since the policy holder has bought the insurance contract.

As we can see the mortality is lower for new entrance at age parity with another individual who has bought previously the same contract.

In this way it can be built a new table called selected tables.

An additional observation is that the increment of mortality related to the previous phenomenon is experienced for a limited number of years, namely r. After the r years the mortality turns back to its own path.

The Selected (before r years) and Ultimate tables (after r years) can be represented as follow:

$$q_{[x]} < q_{[x-1,1]} < \ldots < q_{[x-r-1,r-1]} = \bar{q}_{[x]}$$

Chapter 3

Mortality dynamics

The chapter 3 shows how the mortality and as consequence evolves in the time.

As matter of fact as it has been previously introduced, the mortality is decreasing in the time, and in addition to this there are some trend that can be highlighted.

Given the behavior of the mortality, actuaries have found different methodologies to project and find the distribution of the mortality in the future, as it will be showed in this chapter.

These future estimates are very important for life insurance companies in the case of selling product that supposes cashflows which has long duration in the time (i.e. annuities), for this reason, since pension are basically annuities related product this kind of calculation are extremely important for the pension funds.

3.1 Mortality trends

During the history, it has been observed a change in the longevity and life expectancy of people.

In fact, at the early stage of human life the life expectancy at birth was around 20 or 30 years as has been identified by using fossil and human remains. Others evidence have been provided from data, indeed, since 1870 the Northern European countries started to collect data about life. During that time the life expectancy at birth was about 30-35 years, to increase till 40-45 by the mid-1800s.

A continuous improvement has been observed then, in fact by the middle 20th century life expectancy at birth was about 60-65 years. Finally, to reach 70 years at beginning of 21st century.

An important thing to notice is that the improving in life expectancy has been higher in the last 150 years with respect to the 10000 years before. This can be due to the important scientific discoveries and the improvement of life condition and technologies experienced in the recent past.

There are several factors that have determined the longevity improvement, from health and nutrition to medicine and scientific discoveries.

During the first half of 20th century there have been experience a significant improvement of longevity of infants and children; this progress was due to a better public health and nutrition which helped to pull away infectious diseases.

In the middle of 20th century the improvements in life expectancy was related to medical factor, these factors have improved the life expectancy more related with older ages.

Given these facts, industrialized countries in the world are experiencing a demographical transaction that is leading to an older population. This transaction is caused by sanitary transaction, which has been previously explained, and reproductive transaction, which means that people give rise to less children than before, in fact newborn are in mean less than two for each couple.

The major part of the industrialized country have a pay as you go pension scheme, and since this method is based on the generational exchange, the country are experiencing a lot of difficulties to maintain the actuarial balance between benefit, beneficiaries and pension, pensioners.

For these reasons, the theme of trends in mortality is a very important and essential them in Pension Funds management.

3.1.1 Looking at the data

When approaching mortality forecast it is important to see the plot of the mortality feature in different time period.

In the figure 3.1 it's showed the logarithm of the force of mortality of Belgium male population during different period covering a time span of 120 years.

It can be clearly highlighted the improvement in longevity during the 20th century, as can be seen by the decreasing of the curve of $\log(\mu_x)$ in the time.

The reduction of infectious diseases has led to the greatest improvement, in fact in the young ages the log of the force of mortality has reduced the most in the time.

Another important phenomenon is the hump in the mortality for people aged 18-25, this is mainly due to the circumstance of accidental mortality, accidents, injuries, suicides.

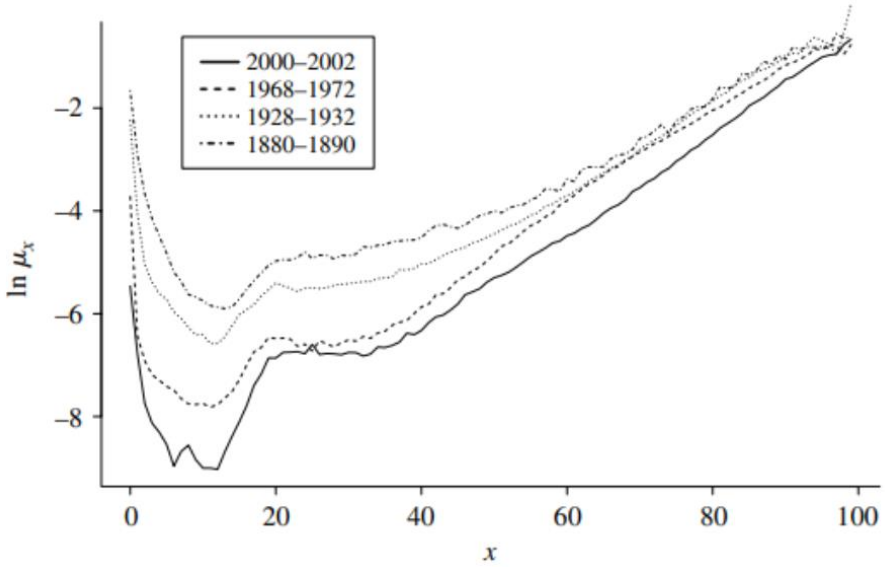

Figure 3.1: Logarithm of μ_x with respect to the age for male Belgium population in the time span 1880-2002. - SOURCE: Modelling Longevity dynamics for pensions and annuities business - Annamaria Olivieri, Ermanno Pitacco, Steven Haberman, Michel Donuit

All these features have been highlighted in some different European countries, not only for that specific population, as it has been showed by *Kannisto et al (1994)*. The paper of the professor has reported an acceleration in the rate of decrease mortality rates at ages over 80 in an analysis of mortality rates.

An additional improvement to the analysis can be performed by looking at the mortality surface, a plot which has three dimensions: $\log(\mu_x)$, age x, year t.

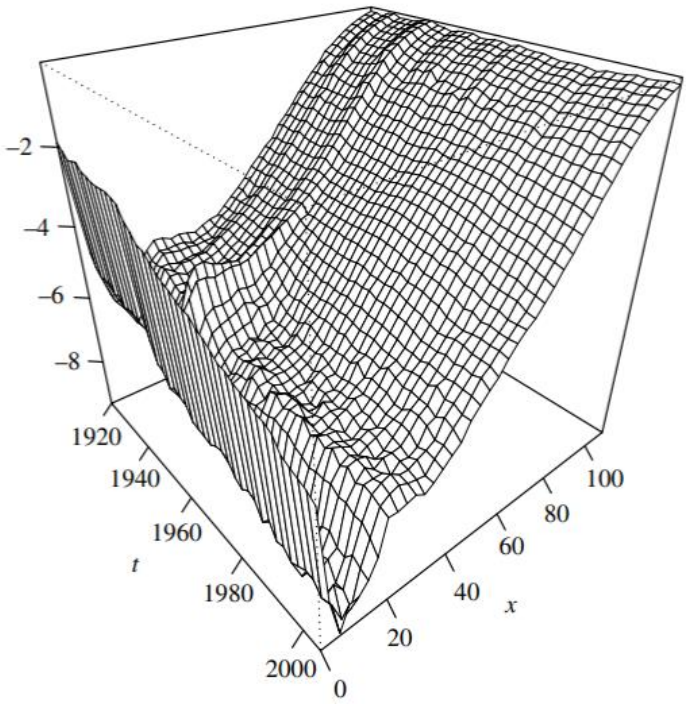

Figure 3.2: Mortality surface for male population in Belgium, between ages 1920 – 2002 - SOURCE: Modelling Longevity dynamics for pensions and annuities business - Annamaria Olivieri, Ermanno Pitacco, Steven Haberman, Michel Donuit

As it is showed in the figure 3.2, it can be noticed that fixing the time t, the $\log(\mu_x)$ is high at the birth, in particular during the beginning of the 1900, this was due to the infant mortality, which can be seen to decrease in future time such as t=2000.

Another phenomenon that can be observed is the mortality hump, which is the difference in $\log(\mu_x)$ between the newborn mortality and the mortality at age 18-25, namely it's higher the delta in the year 2000 with respect to the previous years (i.e. t < 1950).

Historical events such as the World War II has influenced negatively the mortality. In fact, it can be denoted that for the years around that event there have been a strong increase in the mortality. Epidemics, harvest, summer heat waves are such events that affects the mortality, in particular, in the current years it could be interesting to see the effect of the COVID 19 epidemics to the mortality, since it has affected mainly people aged more than 65-70, we could observe an increase in the curve for the years 2020-2021 for that ages.

The death rates displayed for ages bigger than 80 appears very smoothed, this is due to the smoothing procedure that have been used to create that data in the HMD.

3.1.2 Rectangularization and expansion

In the previous chapters it has been stated that the mortality is evolving in the time and we have described in which way has evolved.

Can be found a trend or a particular behavior of the curves in the time? Referring to the figure 3.3, two particular trends in the curves can be observed: they are notoriously called Rectangularization and Expansion.

In the plot is shown the survival function l_x with respect to the attained age x.

It can be clearly seen that the value of l_x is decaying more slowly with respect to the age x, so more people are surviving also at older ages, this phenomenon is called Rectangularization.

In addition to this, it can be seen that the value of the age in which l_x reach the 0 is much higher going forward into the time. It means that more people survive until older ages, this phenomenon is called expansion.

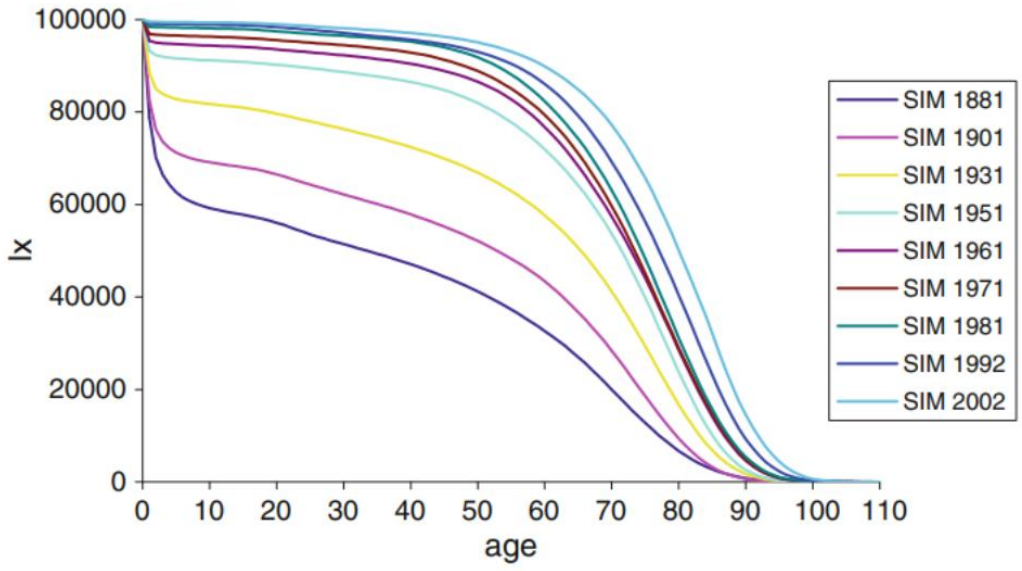

Figure 3.3: Representation of l_x with respect to x for the Italian male population of an interval of years between 1881 and 2002 - SOURCE: Modelling Longevity dynamics for pensions and annuities business - Annamaria Olivieri, Ermanno Pitacco, Steven Haberman, Michel Donuit

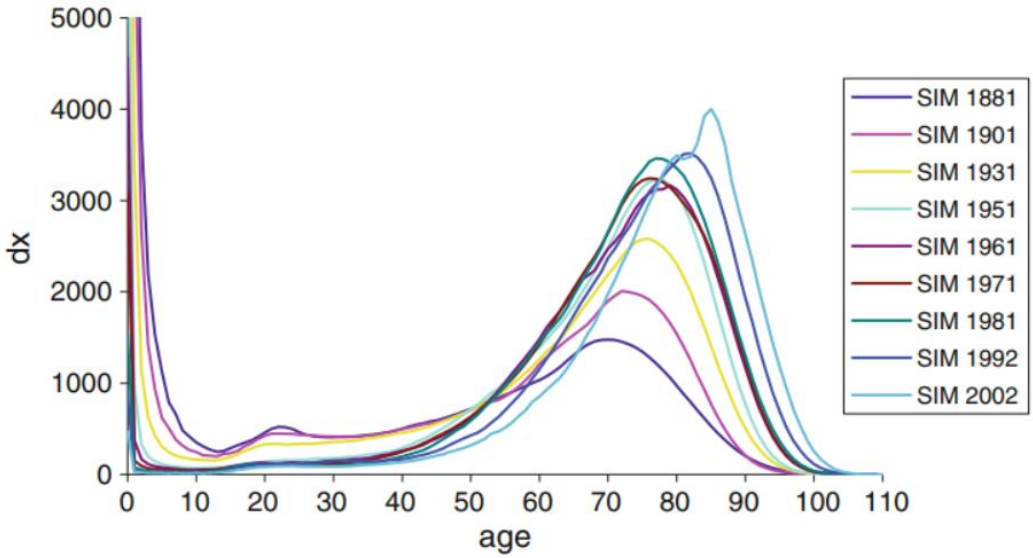

Figure 3.4: Representation of d_x with respect to x for the Italian male population of an interval of years between 1881 and 2002 - SOURCE: Modelling Longevity dynamics for pensions and annuities business - Annamaria Olivieri, Ermanno Pitacco, Steven Haberman, Michel Donuit

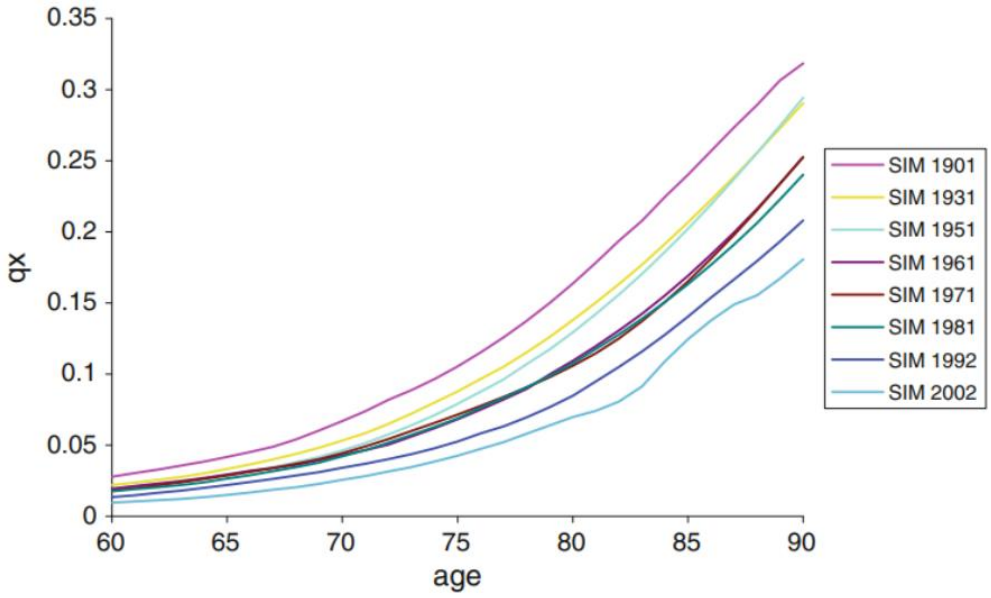

Figure 3.5: Representation of q_x with respect to x for the Italian male population of an interval of years between 1881 and 2002 - SOURCE: Modelling Longevity dynamics for pensions and annuities business - Annamaria Olivieri, Ermanno Pitacco, Steven Haberman, Michel Donuit

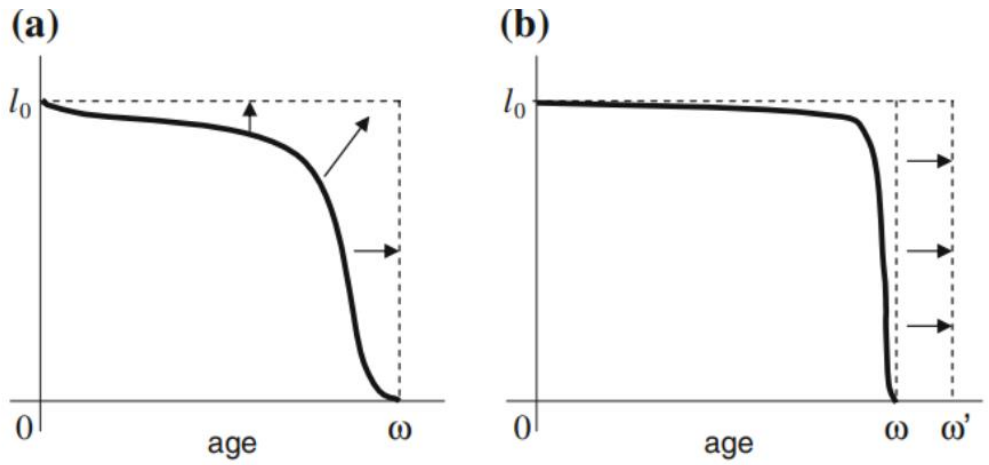

Figure 3.6: Representation of the Rectangularization (a) and expansion (b) phenomena - SOURCE: Modelling Longevity dynamics for pensions and annuities business - Annamaria Olivieri, Ermanno Pitacco, Steven Haberman, Michel Donuit

What have been explained previously on rectangularization and expansion can be clearly seen in figure 3.6.

From these figures can be clearly pointed out that:

1. An increase in life expectancy in general, both for newborn and old ages
2. General decrease in mortality in general (in particular for: newborn, adult and oldest age)
3. Increasing of deaths around the mode at old age (rectangularization)
4. The mode of curve of deaths movers toward old age and is expanding (expansion)
5. Larger dispersion and level of accidental deaths around young ages

3.1.3 Representing Mortality in a Dynamic Context

Mortality is evolving during the time, this fact is very crucial and important since public retirement system and private life annuity business have to take into account this factor.

For this reason, it is important to study this phenomenon, and in order to approach this topic actuaries should move from static mortality tables to a sort of dynamic ones.

The basic idea is to express mortality as function of time of the calendar year, denoted as t. In this way we will have a sort of mortality table for each calendar year t.

And as it's usual to do, also the ages of the individual are needed, so we will consider also x.

Focusing in particular on the one year probability to die, that is defined as $q_x(t)$, that is the one year probability to die for an individual aged x witch are calculated for the year t. This notion is represented in table 3.1.

There are three way to read the probabilities expressed in table 3.1:

1. Vertical arrangements (i.e. fixing a column)

$$q_x(t), q_{x+1}(t), q_{x+2}(t), \ldots, q_\omega(t)$$

This approach corresponds to reading a sequence of period life table each one referring to an individual who lives in a particular year t.

2. Horizontal arrangements (i.e. fixing a rows)

$$\ldots, q_x(t-1), q_x(t), q_x(t+1), q_x(t+2), \ldots$$

In this way, it can be seen the value of the one year probability of die, for an individual aged x, during the whole time series of the years t, which are collected. This approach is useful to understand the trend and how mortality is evolving.

3. Diagonal arrangement

$$q_x(t), q_{x+1}(t+1), q_{x+2}(t+2), \ldots$$

This arrangement corresponds to a sequence of cohort life tables, each one referring to a cohort born in a given year t.

	...	$t-1$	t	$t+1$	$t+2$	...
0	...	$q_0(t-1)$	$q_0(t)$	$q_0(t+1)$	$q_0(t+2)$	...
1	...	$q_1(t-1)$	$q_1(t)$	$q_1(t+1)$	$q_1(t+2)$	...
2	...	$q_2(t-1)$	$q_2(t)$	$q_2(t+1)$	$q_2(t+2)$	...
...	...	...	...	...	...	...
...	...	...	...	...	...	...
x	...	$q_x(t-1)$	$q_x(t)$	$q_x(t+1)$	$q_x(t+2)$	...
$x+1$	...	$q_{x+1}(t-1)$	$q_{x+1}(t)$	$q_{x+1}(t+1)$	$q_{x+1}(t+2)$	...
...	...	...	...	...	...	...
ω	...	$q_\omega(t-1)$	$q_\omega(t)$	$q_\omega(t+1)$	$q_\omega(t+2)$	...

Table 3.1: Representation of dynamic mortality table - SOURCE: Modelling Longevity dynamics for pensions and annuities business - Annamaria Olivieri, Ermanno Pitacco, Steven Haberman, Michel Donuit

In the case that t_n is the current year (i.e. the most recent year), can be observed that the information before t_n are past year (t_1, t_2, …), while information that comes after t_n are future years(t_n+1, t_n+2, … , t*) as represented in figure*. This is an additional information, since observations which come from the past are actually observed and certain.

Since the aim of our research is to estimate future mortality rates, which are not yet observed, in general can be said that observing and learning from past rates. In this way we extrapolate future from pasts as it is done usually in time series analysis, the result of our job is the so called "projected table". The projected tables are a sub-matrix :

$$\{q_x(t)\} \text{ with } x=0,1,2\ldots\omega \text{ and } t=t_n+1, t_n+2, t_n+3, \ldots, t^*$$

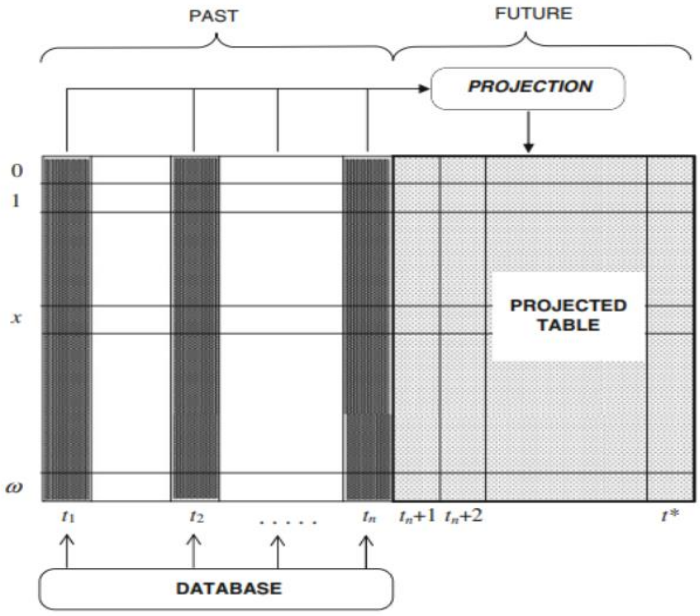

Table 3.2: Representation of projected tables - SOURCE: Modelling Longevity dynamics for pensions and annuities business - Annamaria Olivieri, Ermanno Pitacco, Steven Haberman, Michel Donuit

These kind of tables are very useful in practice, in fact insurance companies that sell long duration of time products (i.e. annuities) need to estimate future mortality rates to be able to price in the best way the product and not risk to underprice or overprice.

3.1.4 Probabilities and Life Expectancies in a Dynamic Context

In this dynamic context also the probabilities of die and to be alive and their relations are changing. Deriving from the diagonal of the table 3.1 the $q_x(t)$, $q_{x+1}(t+1)$, $q_{x+2}(t+2)$, …, that is the cohort life table, can be calculated the annual probability of survive ${}_hp_x(t)$ as:

$$ {}_hp_x(t) = (1 - q_x(t))(1 - q_{x+1}(t+1))(1 - q_{x+2}(t+2)) \dots (1 - q_{x+h-1}(t+h-1)) \quad (\textit{formula } 3.1) $$

This formula simply represents the probability to die for an individual agued x till the age x+h for the calendar year t.

Form the formula 3.1, it can be easily derived the deferred probability to die ${}_{h|1}q_x(t)$ as:

$$ {}_{h|1}q_x(t) = {}_hp_x(t)\, q_{x+h}(t+h) \quad (\textit{formula } 3.2) $$

That is the probability to die for an individual aged x between the age x+h and x+h+1 for the calendar year t.

3.2 Forecasting mortality: different approaches

In this chapter it will be presented the different possible approaches to mortality forecast and how the data are represented in tables in this context.

In particular, there are two main forecasting approaches: deterministic (chapter 3.2.2) and stochastic (chapter 3.2.3).

3.2.1 Preparation of the data

Forecasting mortality can be pursuit with different approaches, but the first and most important thing to be done is to define the dataset, in other words the sample of past observations that will be used for the projection.

A first general methodology in forecasting is the curve fitting and extrapolation technique. The first step of this approach is fitting a curve using a set of past data at

disposal, possibly, by adapting a smoothing effect. The dataset which are used are such that the age x is fixed but the calendar year is changing so that $q_x(t_1)$, $q_x(t_2)$, $q_x(t_3)$, … , $q_x(t_n)$. The curve after the smoothing procedure is indicated as $\psi_x(t_n)$.

Given the fitting procedure then is simple to create an extrapolation of the curve in order to find the future rates, formally we have $q_x(t) = \psi_x(t)$ for all $t>t_n$.

This methodology assumes a horizontal approach in reading time series mortality data, such as in the figure 3.7 and the outcome is $\psi_x(t)$ for each age x in the year t.

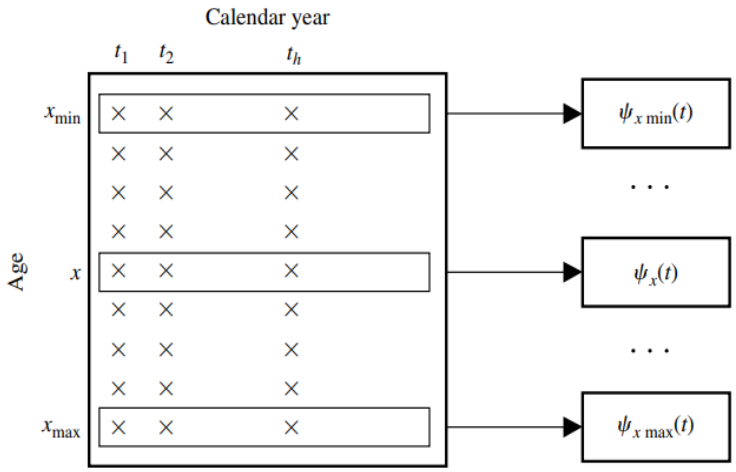

Figure.3.7 The horizontal approach - SOURCE: Modelling Longevity dynamics for pensions and annuities business - Annamaria Olivieri, Ermanno Pitacco, Steven Haberman, Michel Donuit

Performing a projection through this approach means that the extrapolations are performed independently for each q_x, and this means that could arise some inconsistencies regarding the age pattern of mortality.

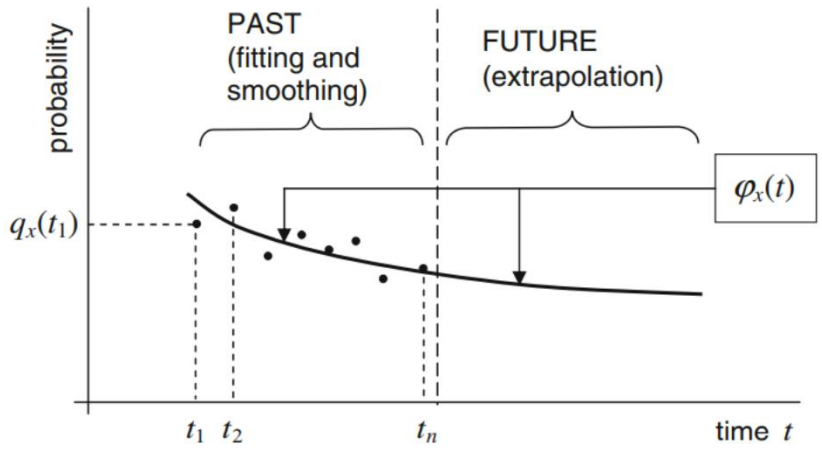

Figure.3.8 Representing the fitting, smoothing and extrapolation procedure - SOURCE: Modelling Longevity dynamics for pensions and annuities business - Annamaria Olivieri, Ermanno Pitacco, Steven Haberman, Michel Donuit

In this contest, data can be considered as simple numbers or they can be considered as a result of a random variable.

When data are considered as pure numbers, the outcome of this research is just a point estimate of future mortality for a given t.

In the case that data are interpreted as an outcome of a random variable regarding the random frequency of deaths, this estimation procedure shall be subjected to more strict statistical assumption and as consequence the future mortality can be represented with point estimate and also interval estimates.

It should be reminded that when projecting the curve, the assumption is that the past trend is continuing in the future. The point is that the time series could be very long, and for this reason, the assumption made above is not satisfied. In this context, it could be necessary to restrict the analysis and the input for the extrapolation in a subset of more recent observation, as in the figure 3.9:

In the figure 3.9 there is a possible outcome of using too long period for interpolating the curve, that is the overestimation of the future mortality.

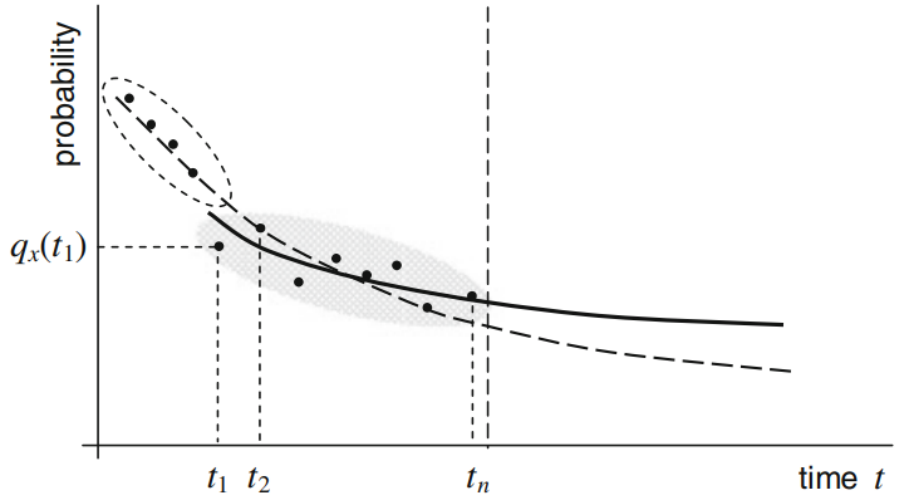

Figure.3.9 Representing the fitting, smoothing and extrapolation procedure - SOURCE: Modelling Longevity dynamics for pensions and annuities business - Annamaria Olivieri, Ermanno Pitacco, Steven Haberman, Michel Donuit

When performing mortality forecast, we can have additional information that may concern a wide range of events, but that aren't represented inside the available data (e.g. Development of medical science).

Additional information can be inserted in the projection by assuming a particular scenario, in this way, more data can be exploited. The problem that arises from this methodology is the arbitrariness of the choice.

In conclusion, there are several methods to project mortality in future, basically in chapter 3.2.2, it will be presented some of the methods that considers the rates as simple numbers and in chapter 3.2.3 data as outcome of a random variable.

3.2.2 Extrapolation via Exponential Models

In this chapter, it will be presented a particular kind of extrapolation, where data are considered as simple numbers.

Let us assume to have the mortality rates for individual aged x where x=0…ω, these rates are available for different calendar year t, where t=0,1, …, n. This can be expressed as:

$$q_0(t), q_1(t), \ldots, q_\omega(t) \text{ with } t=0,1, \ldots, n$$

Suppose that for any age x the trend of the time series of mortality rates q_x can be fitted using an exponential function. In other words, the logarithm of the q_x for each age x is

approximatively linear in the behavior. Further, assume that this trend will continue in future. The mortality can be estimated extrapolating the trend itself. In formal term:

$$\ln q_x(t) - \ln q_x(t') \approx -\delta_x(t-t') \text{ with } t \geq t' \text{ (}\textit{formula 3.3}\text{)}$$

that is equal to:

$$\frac{qx(t')}{qx(t)} \approx e^{-\delta_x(t'-t)}$$

Then by defining the reduction factor $r_x = e^{-\delta_x}$ where r_x is the annual mortality variation factor (reduction factor in the case $r_x<1$) at age x, estimated on the basis of the actual mortality data.

$$q_x(t) \approx q_x(t')r_x^{(t-t')} \text{ (}\textit{formula 3.4}\text{)}$$

In the case that $r_x<1$, it results:

$$\lim_{t\to+\infty} q_x(t) = 0$$

This model (figure 3.10 a) is supposing that going on into the future the mortality is reducing and it's approaching 0 in an exponential way.

This model is only hypothetical and can be substituted with the following model (figure 3.10 b):

$$q_x(t) \approx q_x(t')(\lambda_x + (1-\lambda_x)\, r_x^{(t-t')}) \text{ (}\textit{formula 3.5}\text{)}$$

Then by supposing that $r_x<1$:

$$\lim_{t\to+\infty} q_x(t) = q_x(t')\lambda_x \text{ where } \lambda_x \geq 0 \text{ for each x}$$

This formula is assigning a positive asymptotic mortality, so a mortality that tend to $q_x(t')\lambda_x$.

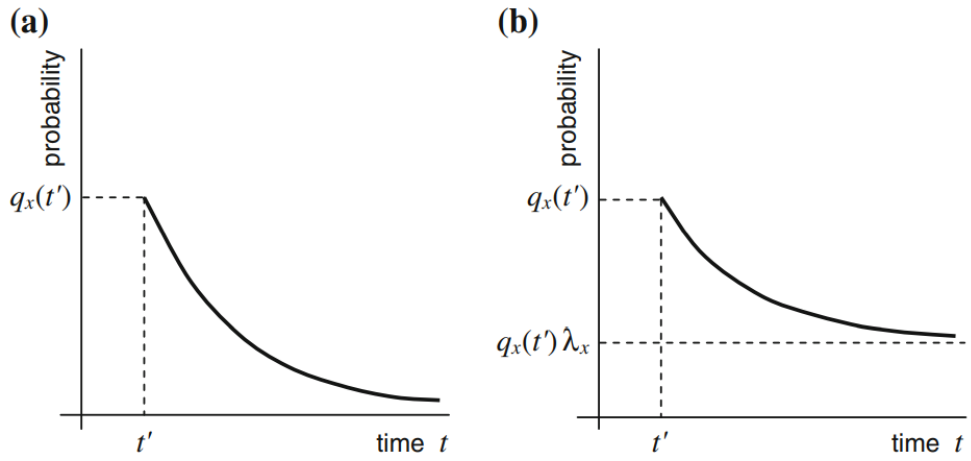

Figure.3.10 Asymptotic mortality in exponential formulae - SOURCE: Modelling Longevity dynamics for pensions and annuities business - Annamaria Olivieri, Ermanno Pitacco, Steven Haberman, Michel Donuit

For what concern the calculation of r_x the least square estimation procedure can be used.

3.2.3 Mortality Projection in a parametric context

When dealing with mortality law, the age-pattern of mortality is summarized by a group of parameters. This feature helps to reduce dramatically the dimension of the of the forecasting problem, since the projection is built only on a few parameters, less degree of freedom, instead of the whole time series of death rates.

Consider for example the Gompertz Law (chapter 2.3), describing the force of mortality as:

$$\mu_x = \varphi(x, \alpha, \beta, \dots)$$

Remembering that it's in a dynamic context, it has to be added the time index t:

$$\mu_x = \varphi(x, \alpha(t), \beta(t), \dots)$$

Let T= $\{t_1, t_2, t_3, \dots, t_n\}$ be the calendar year related to the observations at disposal. Then let a given set of age X, the set of observed values can be represented as:

$$\{\mu_x\}_{x\in X, t\in T} = \{\mu_x(t_1), \mu_x(t_2), \dots, \mu_x(t_n)\}_{x\in X}$$

For each calendar year t_k it can be estimated the parameters to fit the model:

$$\mu_x(t_k) = \varphi(x, \alpha_k, \beta_k, \dots)$$

The parameters can be estimated by using least square, maximum likelihood so that the result is a set of n function of age x, such that:

$$\{\mu_x(t_1), \mu_x(t_2), \dots, \mu_x(t_n)\}$$

The trends in the parameters are calculated via some mathematical formula and hence a set of functions of time t is obtained:

$$\alpha_1, \alpha_2, \dots \alpha_n \rightarrow \alpha(t)$$

$$\beta_1, \beta_2, \dots \beta_n \rightarrow \beta(t)$$

$$\dots$$

As referenced in the figure 3.11:

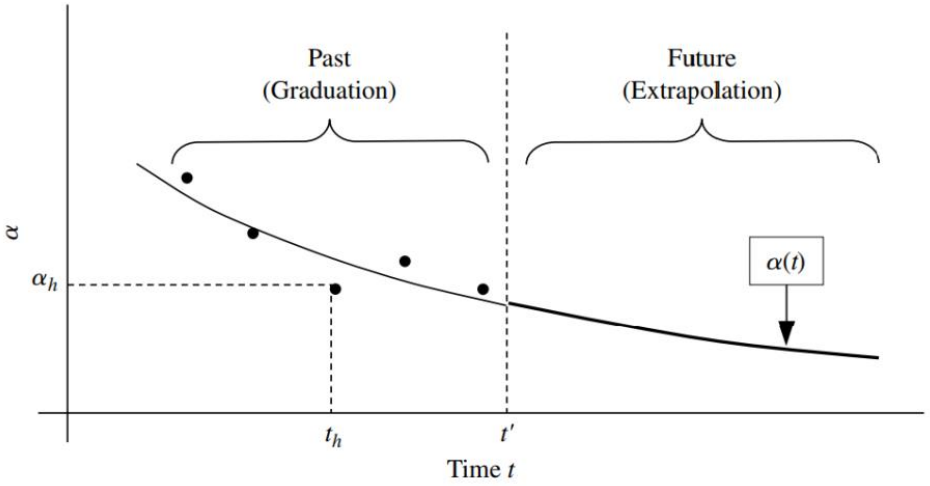

Figure.3.11 Representing the fitting, smoothing and extrapolation procedure - SOURCE: Modelling Longevity dynamics for pensions and annuities business - Annamaria Olivieri, Ermanno Pitacco, Steven Haberman, Michel Donuit

The approach that it has been explained above is the Vertical approach to mortality forecast, since the parameters of the chosen law are estimated for each period table based on observed mortality (figure 3.12).

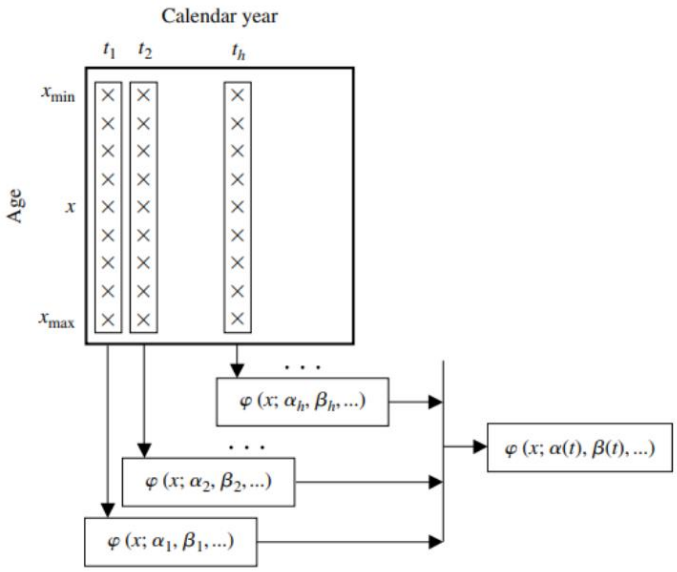

Figure 3.12 Vertical approach - SOURCE: Modelling Longevity dynamics for pensions and annuities business - Annamaria Olivieri, Ermanno Pitacco, Steven Haberman, Michel Donuit

In contrast to the previous approach, the Diagonal approach could be used to estimate the parameters. This procedure uses the cohorts to graduate the parameters of the chosen law, so that the parameters depends on the year at birth τ. A simple representation of this approach can be seen in the following figure 3.13.

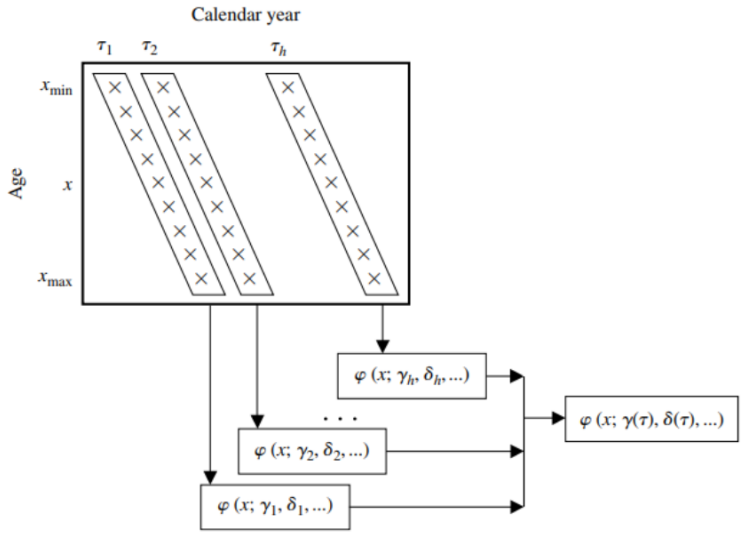

Figure 3.13 Diagonal Approach - SOURCE: Modelling Longevity dynamics for pensions and annuities business - Annamaria Olivieri, Ermanno Pitacco, Steven Haberman, Michel Donuit

3.2.4 Mortality as realization of a random variable

A more interesting approach to mortality forecast is the one which considers the observed mortality as a realization of an underlying random variable.

In particular, two things should be pointed out when using a stochastic approach to forecasting:

- Observed mortality rates are the results of an underlying random variable representing past mortality
- Forecasted mortality rates are estimates of random variables representing future mortality

In addition to this, there are different assumptions to be defined. First of all, a probability distribution of the random number of deaths should be identified and a statistical structure, which links forecast to observations should be specified as in figure 3.14.

The main difference between fitting-extrapolation procedure and the stochastic approach is in the results. In fact, the first methodology produces only a point estimates, this means that the product is just an expectation, without any information about which is the confidence of our expectation.

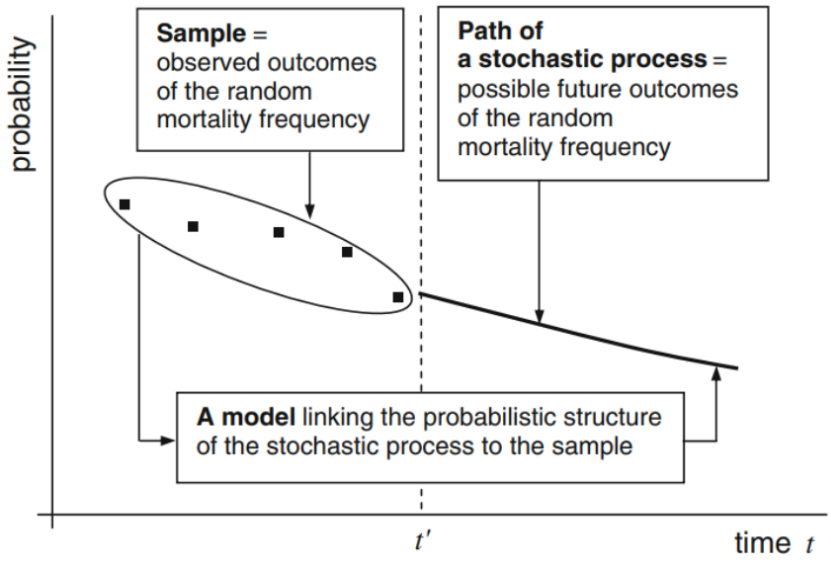

Figure 3.14 A stochastic approach in the fitting–extrapolation procedure - SOURCE: Modelling Longevity dynamics for pensions and annuities business - Annamaria Olivieri, Ermanno Pitacco, Steven Haberman, Michel Donuit

In contrast, when mortality is considered as a realization of an underlying random variable, the product is a point estimate and the interval of confidence, in this way it is possible to assess and measure the risk that the produced forecast is not the real one that will be produced in the future.

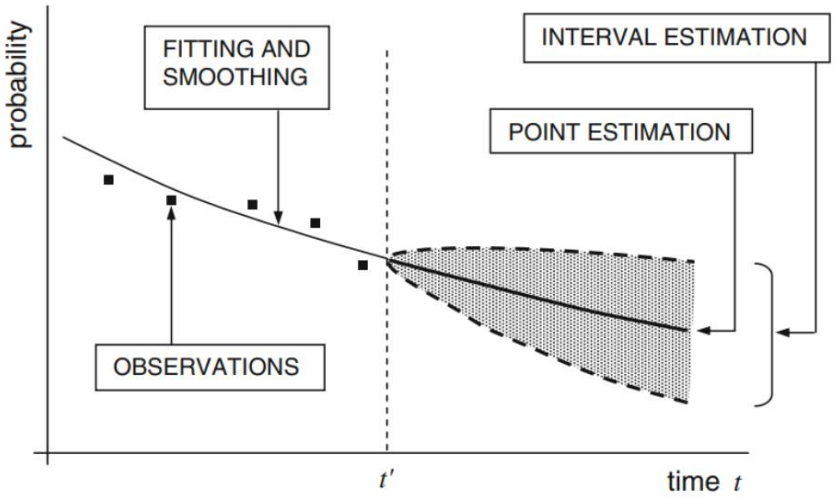

Figure 3.15 Point and Interval Estimation - SOURCE: Modelling Longevity dynamics for pensions and annuities business - Annamaria Olivieri, Ermanno Pitacco, Steven Haberman, Michel Donuit

In this context, there are several models that treat mortality as result of a random variable, one of the most important is the Lee-Carter Method (Lee and Carter 1992, Lee 2000). This approach has been declined in different ways, and the next chapter will be focused on the preparation tools to understand the Lee-Carter model that uses Neural Networks. The last one is an innovative declination and specification of the notorious model and offers relevant and interesting results.

Chapter 4

The Classical Lee-Carter Model

In this chapter it is formulated the Classical Lee-Carter model. This model is one of the classical models used in mortality forecasting. It has been subjected to several improvements and developments: an example is the extension to the multiple population, or the different assumptions made on the distribution of the errors.

Even if the previous models can improve the forecasting ability with respect to the Classical Model, the experiments on this master thesis are developed on the first formulation made in 1992.

In second instance (Chapter 4.2) is described the Box-Jenkins procedure and the ARIMA models which are concerned with the projection in the future of the parameter that is related to the time in the Lee-Carter Model.

4.1 The Classical Lee-Carter Model

In *1992 Ronald D. Lee and Lawrence R. Carter* described for the first time their model using U.S. mortality data over the period 1933-1987. This model consisted in a two phases methodology: a first step is the decomposition of the series, and a second phase consists in the projection of the same previously identified series.

The input of the model is a matrix of age specific mortality rates, one for each year and the output matrix is composed by a forecast of the same matrix in the future years.

The Lee-Carter model defines the natural logarithm of the annual central rate of mortality m_x in the year t and for age x as function of time and the average of the logarithm of the rates.

It can be formalized as:

$$\ln(m_x(t)) = \alpha_x + \beta_x k_t + \varepsilon_{x,t} \text{(formula 4.1)}$$

Where:

α_x is the average age-pattern of mortality over time;

β_x is the deviation from the mortality age pattern over time when k_t varies;

k_t is the univariate index that describes the effect of the time on the changes in the mortality;

$\varepsilon_{x,t}$ is the error term that depends on the year t and the age x;

A first and important hypothesis is that the error term $\varepsilon_{x,t}$ is normally distributed with mean 0 and variance σ^2 constant over the time, witch reflects the age-specific historical influence that is not captured in the model.

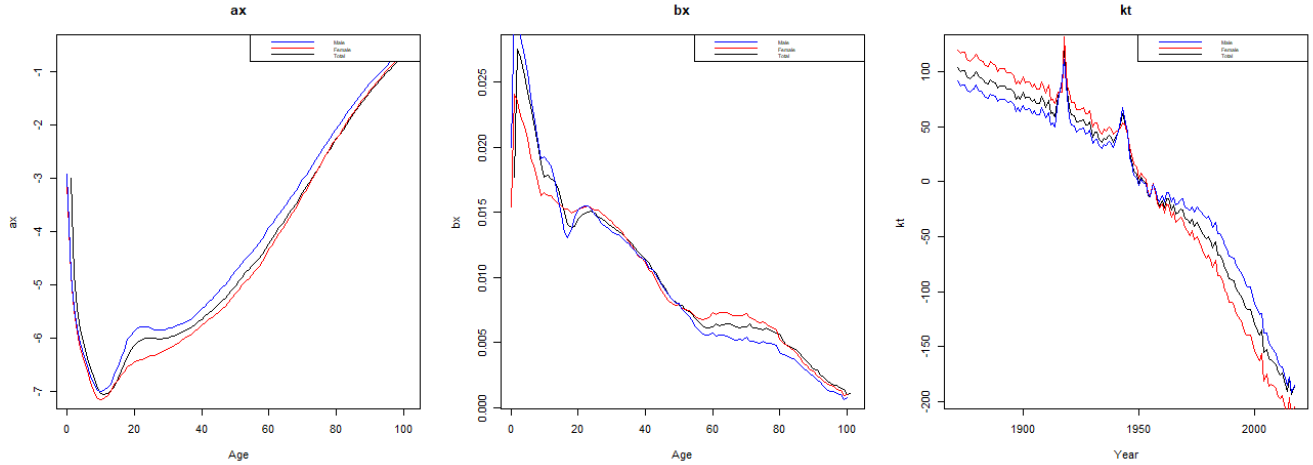

Figure 4.1: The parameters of classical Lee-Carter for the Italian population along the years t: 1872-2017 and for the age between 0-100

An interesting thing to notice from the figure 4.1 is that the k_t is decreasing in the time as general trend, since the mortality is decreasing in the year, but as expected there are two peaks in correspondence of the World Wars. This element should be pointed out in a forecasting analysis, since if the hypothesis is that no more extreme events as the wars will exist in the future, these peaks should be excluded from the analysis.

For what concern the calibration of the parameters, the classical approach is the Singular Value Decomposition, this approach will be explained in the chapter 4.1.2.

4.1.1 Calibration and forecast in the Classical Lee-Carter

In order to estimate the parameters is necessary to have a matrix of rates $m_x(t)$ with $t=t_1, t_2, \ldots, t_n$ and $x=x_1, x_2, \ldots, x_m$.

It's necessary to make some additional constraints: $\sum_{t=t_1}^{t_n} k_t = 0$ and $\sum_{x=x_1}^{x_m} \beta_x = 0$. The model cannot be fit as an ordinary regression model, it adopts the Singular Value Decomposition.

The parameters $\hat{a}_x, \hat{\beta}_x, \hat{k}_t$ are such that to minimize the function:

$$\text{OLS}(\alpha, \beta, k) = \sum_{x=x_1}^{x_m} \sum_{t=t_1}^{t_n} [\ln(\hat{m}_x(t)) - \alpha_x + \beta_x k_t]^2 \text{ (formula 4.2)}$$

where $\hat{m}_x(t)$ is the observed force of mortality.

Computing the first derivative $\frac{\partial \text{OLS}(\alpha,\beta,k)}{\partial \alpha_x}$ and setting it equal to 0, it leads to $\sum_{t=t_1}^{t_n} \ln(\hat{m}_x(t)) = (t_n - t_1 + 1)\,\alpha_x + \beta_x \sum_{t=t_1}^{t_n} k_t$ by the constraint with respect to the time, it can be found:

$$\hat{a}_x = \frac{\sum_{t=t_1}^{t_n} \ln(\hat{m}_x(t))}{(t_n - t_1 + 1)} \text{ for } x=x_1, x_2, \ldots, x_m. \text{ (formula 4.3)}$$

So that $\hat{a}_x$ is the simple average of the $\ln(\hat{m}_x(t))$ with respect to the time.

Then replacing $\hat{a}_x$ with α_x and adding with respect to the age:

$$\sum_{x=x_1}^{x_m} [\ln(\hat{m}_x(t)) - \hat{a}_x] = k_t \sum_{x=x_1}^{x_m} \beta_x + \sum_{x=x_1}^{x_m} \varepsilon_{x,t} \text{ (formula 4.4)}$$

Then, by the constraint:

$$\hat{k}_t = \sum_{x=x_1}^{x_m} [\ln(\hat{m}_x(t)) - \hat{a}_x]$$

By the constraint $\hat{\beta}_x$ is estimable as the slope of the line:

$$\ln(\hat{m}_x(t)) = \hat{a}_x + \beta_x \hat{k}_t \text{ for for } x=x_1, x_2, \ldots, x_m. \text{ (formula 4.5)}$$

In order to calibrate parameters of the LC model, Lee and Carter used Singular Values Decomposition (SVD) of the matrix $Z = \ln(\hat{m}_x(t)) - \hat{a}_x$ to obtain $\hat{\beta}_x$ and $\hat{k}_t$.

Starting from the death rates combined into a matrix:

$$\mathrm{M}=\begin{pmatrix} m_{x_1}(t_1) & \cdots & m_{x_1}(t_n) \\ \vdots & \ddots & \vdots \\ m_{x_m}(t_1) & \cdots & m_{x_m}(t_n) \end{pmatrix}$$

of dimension $(x_m - x_1 + 1)\times(t_n - t_1 + 1)$.

After that the model in formula 4.1 is fitted by Ordinary Least Squares, it's necessary to calculate the SVD of the matrix Z, which is defined as:

$$\mathrm{Z}=\ln(\widehat{M}) - \hat{a}_x=\begin{pmatrix} \ln(\hat{m}_{x_1}(t_1)) - \hat{a}_{x_1} & \cdots & \ln(\hat{m}_{x_1}(t_n)) - \hat{a}_{x_1} \\ \vdots & \ddots & \vdots \\ \ln(\hat{m}_{x_m}(t_1)) - \hat{a}_{x_m} & \cdots & \ln(\hat{m}_{x_n}(t_n)) - \hat{a}_{x_m} \end{pmatrix}$$

of dimension $(x_m - x_1 + 1)\times(t_n - t_1 + 1)$.

Now $\hat{\beta}_x$ and $\hat{k}_t$ are such that:

$$\text{minimize } \mathrm{O_{LS}}(\beta, k) = \sum_{x=x_1}^{x_m} \sum_{t=t_1}^{t_n} [z_{xt} - \alpha_x + \beta_x k_t]^2 \text{ (formula 4.6)}$$

Let u_1 be the eigenvector corresponding to the largest eigenvalue of Z^TZ. Let v_1 be the corresponding eigenvector of ZZ^T. The best approximation of Z in the least-squares sense is known to be:

$$\mathrm{Z}\approx Z^* = \sqrt{\lambda_1}\, v_1 u_1^T$$

From which it can be found:

$$\widehat{\beta}=\frac{v_1}{\sum_{j=1}^{x_m-x_1+1} v_{1j}} \text{ (formula 4.7)}$$

and

$$\widehat{k}=\sqrt{\lambda_1}\left(\sum_{j=1}^{x_m-x_1+1} v_{1j}\right)u_1 \text{ (formula 4.8)}$$

Given that $\sum_{j=1}^{x_m-x_1+1} v_{1j} \neq 0$.

After that all the parameters are calibrated and calculated in the classical way, it's possible to make the projection of the mortality table.

The projection consists in projecting k_t in the future (i.e. where $t>t_n$). The classical formulation of the LC model adopts the ARIMA model as the propter one for the projection, this kind of model applied to this methodology will be described in chapter 4.2.

The R software contains several packages for mortality modelling and demography.

In order to obtain the mortality tables, it's possible to use the hmd.mx () function from the package demography by Rob Hindman. This function allows to directly load in R, the data about the mortality of the indicated country.

From the previously defined package, it's very useful the lca () function, it allows to compute the Lee-Carter model through the Singular Value Decomposition.

The Classical Lee-Carter method suffers different problems, as evidenced by Lee (2000). Some of them can be summed up in the following points:

- The time series of k_t that is observed in the fitting period can be different from the one of the whole historical experience and eventually cannot reflect a fundamental property of mortality change over time.
- The method estimates a certain pattern of change in the age distribution of mortality, such that the rates of decline at different ages (given by b_x (dk_t/dt)) always maintain the same ratios to one another over time. But in practice, the relative speed of decline at different ages may vary. (Lee, 2000)
- The method is not able to insert exogenous information about future trends.
- The method doesn't account for the cohort effect.

For these reasons, different extensions of the model have been proposed, both by *Lee (2000)* and by other authors. A certain kind extension regards the parameters decomposition, that can be done through the Weighted Least Squares as suggested by *Wilmoth (1993)*, the Maximum Likelihood Estimation as proposed by *Brouhns et al. (2002)*. Other models have extended the Lee-Carter model introducing additional effects such as a cohort dependent component that is not considered in the Classical Lee-Carter, such as *Renshaw and Haberman (2006)*. Additional models extend the methodology to multiple populations as *Lee et al. (2005)* and *Kleinow (2015)*. Another influential model is the two factors approach to mortality that has been proposed by *Cairns, Blake and Dowd (2006)*.

4.2 Box-Jenkins technique and ARIMA Model

In the first formulation of the Lee-Carter Model, the authors projected the value k_t by applying an ARIMA (0,1,0) (i.e. random walk with drift) for modeling the mortality index for US population.

In the practice the Box-Jenkins technique, and in particular, ARIMA are the preferred methods to forecast the k_t, anyway not always the best model is represented by a random walk with drift but it depends by the time series.

The Box-Jenkins procedure consists in method for identifying, preparing, checking, fitting using an ARIMA model a time series. This analysis consists in three steps: Identification, Estimation, Diagnostic Checking.

The identification concerns all the steps taken to identify the kind of time series, such as the plot for autocorrelation and partial autocorrelation, the plot of the time series and all what is needed to define the parameters p, d, q for the ARIMA model.

The step of the Estimation is based on the previously defined ARIMA (p,d,q) defined in the previous stage, using the Maximum Likelihood.

Finally, the Checking procedure consists in studying the residuals, whether autocorrelation is present or not. The model can be considered good in case autocorrelation and partial autocorrelation are low.

ARIMA is the acronym for Auto-regressive (AR) Integrated (I) Moving-Average (MA) models, this means that are composed by different parts that will be explained in this chapter.

In order to explain the how the method works is necessary to describe what is an Auto Regressive (AR) process and a Moving Average process (MA), witch in practice compose an ARIMA Process.

The process $\{K_t, t \in Z\}$ is an Auto Regressive process of order p, denoted with AR(p) if:

$$K_t = \phi_1 K_{t-1} + \phi_2 K_{t-2} + \cdots + \phi_p K_{t-p} + u_t \text{ (formula 4.9)}$$

Where K_t is stationary, $u_t \sim WN(0, \sigma_u^2)$ and $\phi_1, \phi_2, \dots, \phi_p$ are constants.

In practice an Auto Regressive model is based on the idea that the current value of the series, K_t, can be explained as function of p past values, where p determine the number of steps into the past needed to forecast the current value.

The process $\{K_t, t \in Z\}$ is a Moving Average process of order q, denoted with MA(q) if:

$$K_t = u_t + \theta_1 u_{t-1} + \theta_2 u_2 \dots + \theta_q u_{t-q} \text{ (formula 4.10)}$$

Where $u_t \sim WN(0, \sigma_u^2)$ and $\theta_1, \theta_2, \dots, \theta_q$ are constants.

Given these processes, it' possible to define the process $\{K_t, t \in Z\}$ as an Auto Regressive Moving average process of order p, q (ARMA(p,q)) as:

$$K_t = \phi_1 K_{t-1} + \phi_2 K_{t-2} + \cdots + \phi_p K_{t-p} + u_t + \theta_1 u_{t-1} + \theta_2 u_2 \dots + \theta_q u_{t-q} \text{ (formula 4.11)}$$

For all t ∈ Z , $u_t \sim WN(0, \sigma_u^2)$ and $\phi_1, \phi_2, \dots, \phi_p, \theta_1, \theta_2, \dots, \theta_q$ are constants.

Finally, the ARIMA (p,q,d) is defined as an ARMA(p,q) where the difference of order d (∇^d)is taken in order to remove the non-stationarity in the data, such as:

$$\nabla^d \mathrm{K_t} = \phi_1 \nabla^d K_{t-1} + \phi_2 \nabla^d K_{t-2} + \cdots + \phi_p \nabla^d K_{t-p} + u_t + \theta_1 u_{t-1} + \theta_2 u_2 \dots + \theta_q u_{t-q}$$

(formula 4.12)

For all t ∈ Z , $u_t \sim WN(0, \sigma_u^2)$ and $\phi_1, \phi_2, \dots, \phi_p, \theta_1, \theta_2, \dots, \theta_q$ are constants and $\nabla^d$is the differencing operator of order d.

The main task when using ARIMA model is the choice of the parameters p, q, d of the model. Generally, the selected model is the one that minimize the AIC (Akaike Information Criteria) or the BIC (Bayes Information Criteria) that are defined as:

$$\mathrm{AIC(p,q)} = \log(\hat{\sigma}^2_{u,p,d}) + \frac{T+2(p+q)}{T} \text{ (formula 4.13)}$$

Where $\hat{\sigma}^2_{u,p,d}$ represents the maximum likelihood estimated variance of the white noise, the T is the number of observations.

$$\mathrm{BIC(p,q)} = \log(\hat{\sigma}^2_{u,p,d}) + (p+q)\frac{\log T}{T} \text{ (formula 4.14)}$$

An important hypothesis when dealing with time series in ARIMA contest is the stationarity.

The stochastic process that is underlying the time series is stationary when the statistical features of the process don't vary over time. In practice there are two forms of stationarity:

1. Strong stationarity: the probability distribution is invariant shifting in the time
2. Weak stationarity: the mean and the variance are not changing in shifting in the time and the covariance between the values at any two points, t and t+k, depends only on the difference k between the two times

When applying ARIMA models the time series should be stationary, if this doesn't happen is necessary to take a d order differencing in order to make the series stationary, then is possible to proceed to choose the orders p, q such that reflects the best behavior of the time series.

In order to find the best parameters p,q it could be interesting to examine the autocorrelation function (ACF) and the partial autocorrelation function (PACF) starting from the definition of the autocovariance function.

The autocovariance function of the stochastic process $\{K_t; t \in Z\}$ is defined as:

$$\gamma_k(t,s) = Cov(K_t, K_s) = \mathrm{E}[(K_t - \mu_{kt})(K_s - \mu_{ks})] \text{ (formula 4.15)}$$

Where $\mu_{kt} = E(K_t)$ is the mean function of $\{K_t; t \in Z\}$ which describes the expectation of the random variable over the time.

This indicator measures the strength of the linear relationship between the random variables $K_t \; and \; K_s$.

In the particular case where s=t, it's possible to define the variance, such as:

$$\gamma_k(t,t) = Cov(K_t, K_t) = \mathrm{E}[(K_t - \mu_{kt})(K_t - \mu_{kt})] = \mathrm{E}[(K_t - \mu_{kt})^2] = \mathrm{Var}(K_t) \text{ (formula 4.16)}$$

The autocovariance function is an important indicator, otherwise it depends on the unit of measure, for this reason a more useful indicator is the autocorrelation function which measures the strength of the relationship between the random variables but not depend on the dimension.

The autocorrelation function of the weakly stationary stochastic process $\{K_t; t \in Z\}$ is defined as:

$$\rho_k(h) = \frac{\gamma_k(h)}{\gamma_k(0)} \text{ (formula 4.17)}$$

Since it measures the correlation, the indicator is defined between +1 and -1.

Let K, Y, Z be random variables, then regress K on Z and Y on Z, the partial autocorrelation function is defined as:

$$\rho_k(h) = Corr(K - \hat{K}, Y - \hat{Y}) \text{ (formula 4.18)}$$

This indicator gives the partial correlation of the time series with its own lagged values.

In the figure 4.2 is possible to see an example of ACF and PACF of a time series.

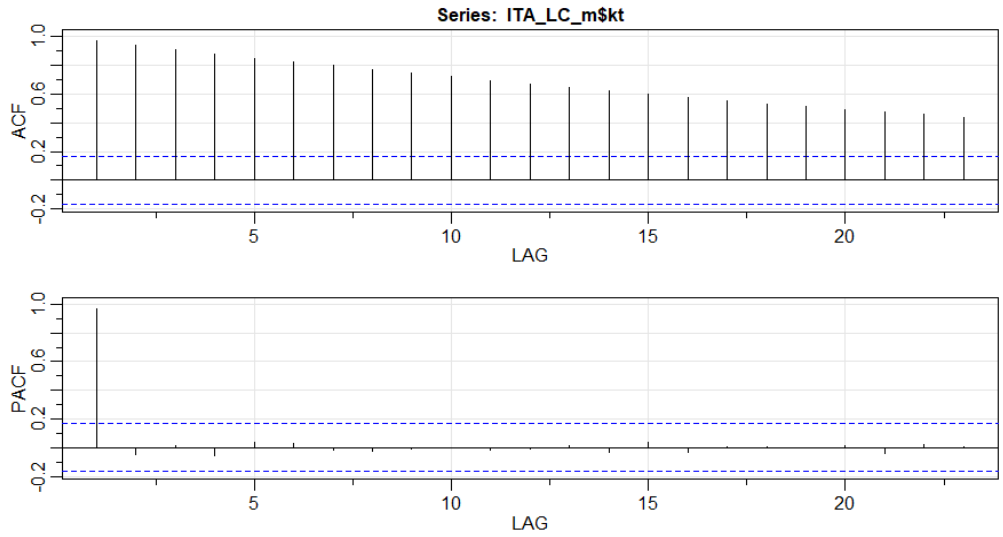

Figure 4.2: ACF and PACF of K_t in Italian male population calculated on data between years 1921 and 2018

Form the ACF in the figure 4.1 is easily to see that there is a linear decay, this supports the hypothesis of non-stationarity of the series, so that a transformation is needed, in addition looking at the PACF is interesting to notice that the lag of value one of K_t, in other word K_{t-1}, is significant in explaining the value of K_t.

The last phase of the Box-Jenkins technique is the Checking of the results. In this step is important to inspect the residuals.

The residuals are determined as the difference between the fitted values, in other words values that comes from the estimation $\hat{k}_t$ based on the model and the real values k_t.

$$e_t = k_t - \hat{k}_t \text{ (formula 4.19)}$$

Residuals are very useful in checking whether the model has correctly captured the information that comes from the data. When the model is correctly specified, they should have the following properties:

1. Residuals are uncorrelated. In the case that residuals are correlated it means that an important information that can be used to predict the future behavior has not been captured in the model.
2. Residual have mean equal to 0. In the case that residuals are not 0 in mean it means that the results obtained are biased.
3. Residuals have constant variance.
4. Residuals are normally distributed.

These firsts two hypothesis are very important to be satisfied in order to have a good model, while the last two are good property but not mandatory. Another way to specify these assumptions is that residuals should be a realization of a white noise.

There are several tests that can be done in order to assess the adequacy of the residuals. One of the most used is the Box-Pierce test.

The test is based on the following statistic:

$$Q_H = T \sum_{h=1}^{H} \hat{\rho}_k(h)^2 \quad \text{(formula 4.20)}$$

Where T is the number of the observations, H is the maximum lag being considered and $\hat{\rho}_x(h)$ is the autocorrelation at lag h. When the number of observation T is large the statistics follows a chi-square distribution with H-p-q degrees of freedom.

Another related statistic is the Ljung-Box test, which is defined as:

$$Q^* = T(T+2)\sum_{r=1}^{H}(T-r)^{-1}\ \hat{\rho}_k(r) \quad \text{(formula 4.21)}$$

Large values of Q^* suggest that the autocorrelations do not come from a white noise series.

Both methods for large p-value rejects the null hypothesis witch states that residuals not come from a white noise distribution.

The last important point is the evaluation of forecast accuracy. In order to reach reliable results, it's needed to not evaluate the performance on the data that are used to build the model (i.e. training data).

In facts the first thing to do is to divide the time series in training and testing data as in the following figure:

Figure 4.3: Division of the time series in test and training data (Source: Forecasting: Principles and Practice – Hyndman R.J., Athanasopoulos G. - Monash University, Australia - 2018)

As it can be seen in the figure 4.3, the time series is subdivided in training and test data in a ordered manner, this is because the time series data are time-ordered data and for this reason is interesting to preserve the order.

The training data are composed by a sample of the whole population that is used to build the model and calibrate parameters, in contrast the test data are the part of the series

that is not used in the model calibration and it's used only to evaluate the performance of the model.

Usually test and training part are subdivided according to percentage, such as 20% test , 80% training; anyway there are cases in which the series is very short and it's very difficult to build the model on 80% training data and so the dimension of test can be reduced or eliminated.

The forecast error is evaluated using the test data, and can be defined as the difference between the observed values and the forecast, such as:

$$e_{T+h=} k_{T+h} - \hat{k}_{T+h} \text{ (formula 4.22)}$$

where training data are composed by $\{k_1, \dots, k_T\}$ and test data as $\{k_{T+1}, k_{T+2}, \dots\}$.

It's possible to measure forecast accuracy by calculating the error in different ways.

Two of the most common errors measurement are the Mean Absolute Error (MAE) and the Root Mean Squared Error (RMSE), defined as follows.

$$\text{MAE} = \text{mean}(|e_t|) \text{ (formula 4.23)}$$

$$\text{RMSE} = \sqrt{\text{mean}(e_t)^2} \text{ (formula 4.24)}$$

Both measures are scale dependent, so are useful only in case there are compared different series with the same units.

4.2.1 Arima Modelling in R and SAS®

The ARIMA model can be fit using different software.

When dealing with SAS® software it's possible to use the proc arima specifying the target variable, and the time order, this procedure directly proposes in the output some useful information about the residuals.

In order to check residuals also in R there are some function, such as checkresiduals () from forecast package by Rob J Hyndman. This function plots the residuals, the ACF of the residuals, fit a normal curve on the residuals and also it computes a Ljung-Box test for residuals checking.

R software has a function called auto.arima () which can be useful to find the best model according to the AIC/BIC.

The output of this function is the result of the following algorithm:

Hyndman-Khandakar algorithm for automatic ARIMA modelling

1. The number of differences $0 \leq d \leq 2$ is determined using repeated KPSS tests.
2. The values of p and q are then chosen by minimising the AICc after differencing the data d times. Rather than considering every possible combination of p and q, the algorithm uses a stepwise search to traverse the model space.
   a. Four initial models are fitted:
      - ARIMA$(0, d, 0)$,
      - ARIMA$(2, d, 2)$,
      - ARIMA$(1, d, 0)$,
      - ARIMA$(0, d, 1)$.

      A constant is included unless $d = 2$. If $d \leq 1$, an additional model is also fitted:
      - ARIMA$(0, d, 0)$ without a constant.

   b. The best model (with the smallest AICc value) fitted in step (a) is set to be the "current model".

   c. Variations on the current model are considered:
      - vary p and/or q from the current model by ± 1;
      - include/exclude c from the current model.

      The best model considered so far (either the current model or one of these variations) becomes the new current model.

   d. Repeat Step 2(c) until no lower AICc can be found.

(Source: Forecasting: Principles and Practice – Hyndman R.J., Athanasopoulos G. - Monash University, Australia -2018)

This algorithm is performing an analysis which is similar to the Box and Jenkins procedure, the only thing that is not covered is the analysis of the residuals arising from the model.

Chapter 5

Neural Network Theory

In the chapter 5 is explained the theory underlying the Neural Network models, in particular, Recurrent Neural Network.

Here there are three chapters, the first about the Feed Forward Neural Network, the second about the Recurrent Neural Network, and the third one about the hyperparameters that are present when tuning this kind of architecture.

This chapter is useful to prepare to the application that has been conduct in the chapter 6.3 about Long-Short Term Memory and Gated Recurrent Units Neural Networks.

5.1 Artificial Neural Network

The idea of Neural Network has been developed for the first time by *McCulloch and Pitts* in the paper named “A logical calculus of ideas imminent in network activity”.

The name of the technique recalls the neuron, which is the unit of cognition. In fact, in the biological neuron, the input arrives through the dendrites and then it reaches the nucleus of the neurons and finally it goes out to the axon terminal. The neuron is the fundamental unit of the brain and the biological brain a lot of characteristics in common to this network of neurons: the ability to learn, the efficiency, the ability to deal with fuzzy information.

The figure 5.1 represents the first developed structure of the Neural Network, as it can be noticed, the architecture is similar to the biological neuron, in fact, the input

(dendrites) pass through the transfer function (nucleus) and then goes into an output (axon terminal).

In particular, the input, $X_1 \ldots X_p$ are multiplied by the weights $W_1 \ldots W_p$, then the output of the sum is passed through a step function which converts the results bigger than 0 to 1 and the sum lower than 0 to -1.

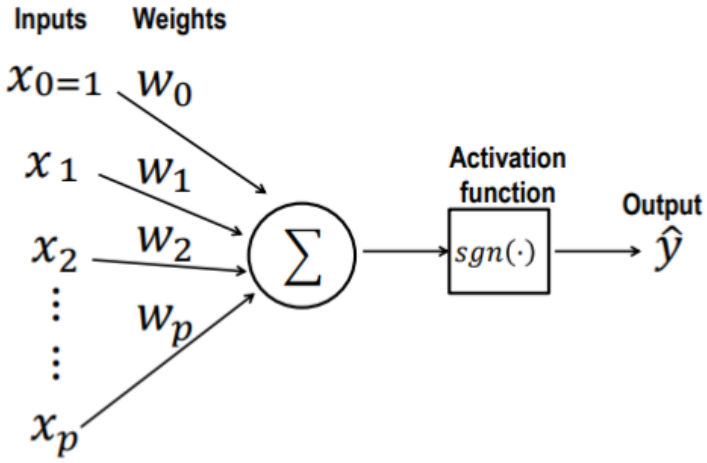

Figure 5.1: McCulloch and Pitts Neural Network representation (SOURCE: Paci L. – Empirical Research Course notes – A.Y. 2019-20)

The equation that represents the output $\hat{y}$ is the following:

$$\hat{y} = sgn\ (w_0 + \sum_{j=1}^{p} w_j x_j) \text{ (formula 5.1)}$$

Where $\hat{y}$ is the binary output, $X_1 \ldots X_p$ the input, $W_1 \ldots W_p$ the weights, W_0 the bias, $sgn(\cdot)$ the function.

This input-output relation can be generalized to the following case:

$$\hat{y} = g\big(\varphi(x, w)\big) \text{ (formula 5.2)}$$

Where $\hat{y}$ represents the output, g is the activation function, φ is a linear function, $X_1 \ldots X_p$ the input, $W_1 \ldots W_p$ the weights, W_0 the bias term.

The neural network designed in the figure 5.1, when generalized, can be seen as the simplest form of a more complex structure. That in some cases, can help solve more complex problems.

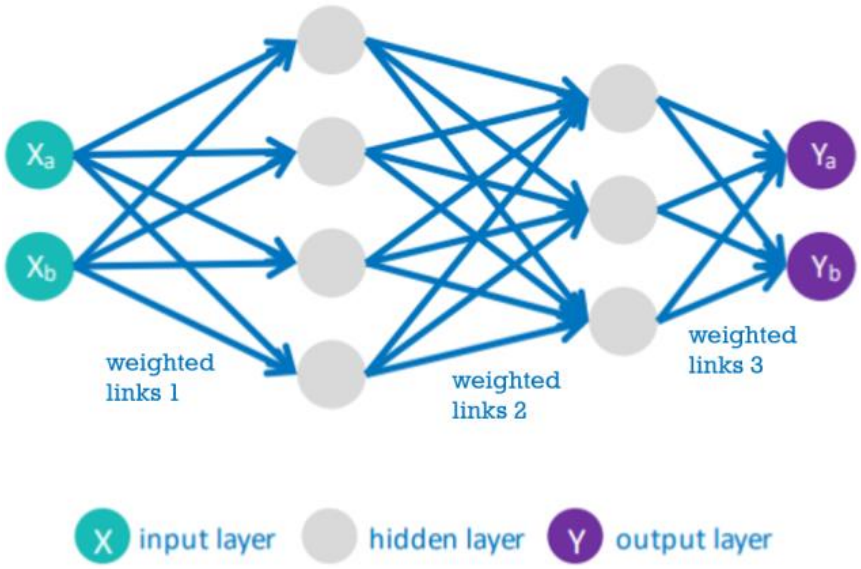

Figure 5.2: Feed Forward Neural Network, Multilayered Perceptron (SOURCE: SAS Institure)

The structure here represented is an Artificial Neural Network of the class Feed Forward Neural Network Multilayer Perceptron.

The structure represents a list of input X_a and X_b, two hidden layers composed by hidden units and an output layer. The total of the layer is 4. The structure works in the following way: the input units are weighted, then the hidden units get the sum of the input and the weight and apply an activation function, which will be explained in the chapter 3, the output of each hidden units goes into the next hidden units to receive the same treatment. Finally, the output units receive the weighted sum of the hidden units and apply an activation function to this sum. In this kind of Network, differently from the Recurrent Neural Network structure, which will be explained in the next chapter, the network is feed in a forward direction, in other words, the information moves in one direction, and never goes backward.

5.2 Recurrent Neural Network

Recurrent Neural Networks (RNNs) are Artificial Neural Networks (ANN) which are useful to model sequential data.

These models have been proposed the first time by *Rumelhart et al. in 1986*, others important developments have been proposed by *Webros in 1988 and Elman in 1990*.

In practice, they are very successfully used in speech analysis, text analysis and in time series application thanks to their ability to manage with the correlation among the observations that characterize sequentially ordered data.

Comparing this structure to feed-forward neural networks (discussed in previous chapter) the main difference is that the flow of information is not only in one direction, from the input to the output. In facts as it can be seen in the figure the data that are feed into the hidden unit and back to themselves. In this way, the information from the previous states, in the sequence, can be saved and kept into account for the forward steps, this creates a sort of memory that keep long term information.

In practice, if we consider the inputs as part of a time series where X_a and X_b are ordered, the error in predicting Y_a is used to adjust the weights used to predict Y_b.

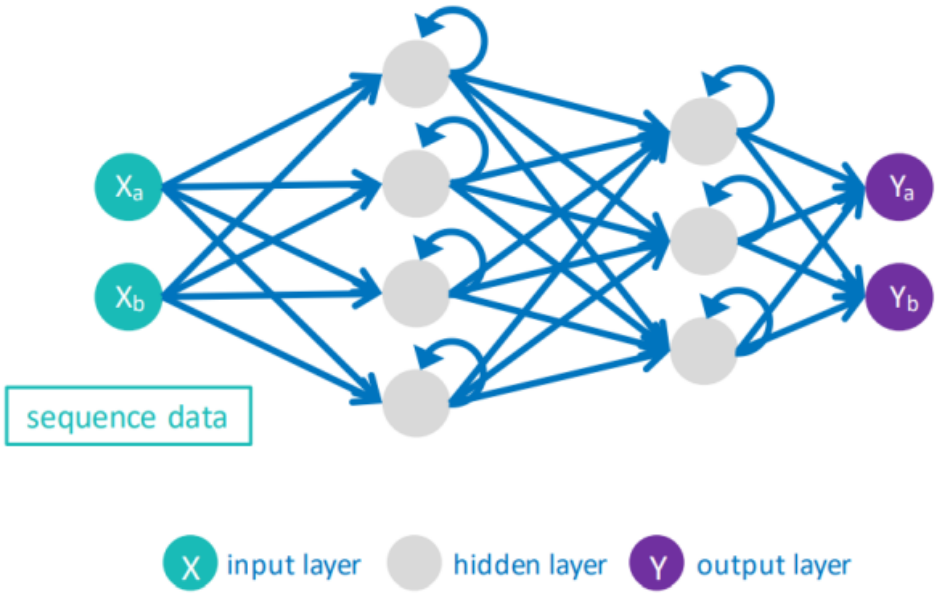

Figure 5.3: Recurrent Neural Network Structure, multilayered (SOURCE: SAS Institure)

The dept in the Recurrent Neural networks is different from the DNN (Deep Neural Network), in fact, if the "deep" of a Neural Networks refers to the presence of a composition of several nonlinear computational layers the RNN can be already considered dept models. This is due to the fact that any RNN can be expressed as a composition of multiple nonlinear layers when unfolded in time, as in the figure 5.4.

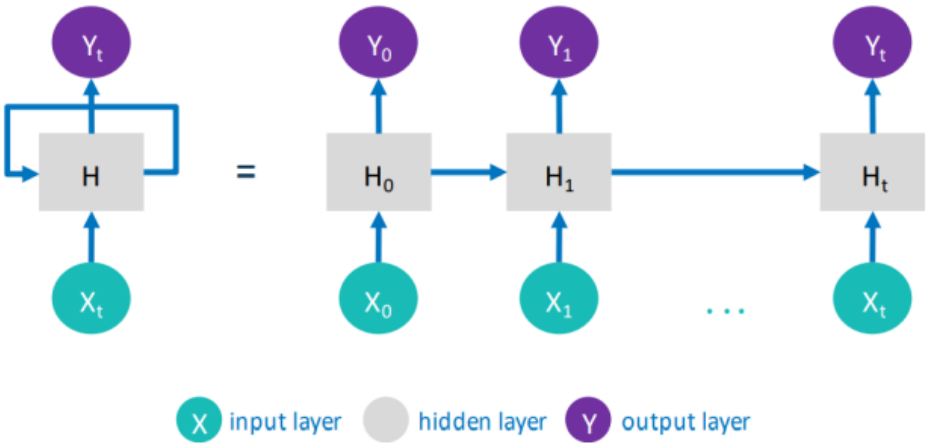

Figure 5.4: Unfolded RNN structure (SOURCE: SAS Institure)

In the figure 5.4 there is represented a RNN with a single input, a single hidden unit and a single layer.

The model can be seen as a chain of networks that learn from the previous input, but witch weights are shared across the whole chain, in a sort of recurrent way.

The unfolded RNN shows that there is one hidden unit per input, and the previous input and hidden unit are combined together to generate the output.

The chain of networks X_0 to X_t represents the information flows across the sequence, where the information from the past is combined with the current input to generate the prediction in the output.

Even if the network can be unfolded, the different time steps the weight inside the whole chain are the shared in common.

By taking into account the RNN that is represented in the figure 5.4, it's possible to develop the equations that represent this simple network.

$$Y_t = sigmoid[W_Y \cdot H_t + b_Y] \text{ (formula 5.3)}$$

$$H_t = \tanh\left[W_H \cdot H_{t-1} + W_X \cdot X_t + b_H\right] \text{ (formula 5.4)}$$

Where:

- W_Y is the hidden to output weight
- W_H is the hidden to hidden weight
- W_X is the input to hidden weight
- Y_t is the output
- H_t is the hidden unit
- X_t is the input

- $b_Y \; and \; b_H$ are the bias term

The formulas represent the recursion that is present in the RNN, in fact, the current hidden unit H_t is function of the previous hidden unit H_{t-1}. The main idea is that by tuning the value of the weight it's adjusted how much the past sequence elements can affect the current prediction.

The training process of the RNN model starts with the computation of the total error, which is simply the sum of the losses of the overall timestamps. The aim of the training is to minimize the result of the objective function which is the total error or loss. For this task the algorithm that is used is the Gradient Descent, in particular, to train a RNN is used the Back Propagation Through Time (BPTT). This algorithm has the peculiarity that the RNN is unfolded and, differently from the Feedforward Propagation, the error is the propagated backward for all the time step, one at time, through the entire input sequence.

There are two main problem when approaching the RNN, exploding and vanishing gradient.

"Exploding gradient refers to the large increase in the norm of the gradient during the training phase. This phenomenon is caused by the explosion of the long-term components, which can grow exponentially more than short term ones" (*Pascanu at al. 2012*).

"The vanishing gradient problem refers to the opposite behavior, when long term components go exponentially fast to norm 0, making it impossible for the model to learn correlation between temporally distant events" (*Pascanu at al. 2012*).

The solution to these problems is to control witch information is necessary and which should be excluded from the network. In fact, RNN can pass irrelevant information forward, because the scheme of the weight requires that all the information are needed and flow through the whole sequence (non-zero weight on the previous hidden unit) or for the past to be irrelevant (weight of zero on the previous hidden unit).

In order to solve these problems subtypes of RNN are used: Long Short-Term Memory, Gated Recurrent Unit.

5.2.1 Gated Recurrent Unit

A Gated Recurrent Unit was proposed by *Cho et al. in 2014* to make each unit to capture dependency in a more flexible way on different time scales (*Cho et al*).

Gated Recurrent Unit (GRU) are a form of RNN architecture that introduces “gates” to prevent some information to pass through the hidden units. In particular, the model uses an update gate u_t and a reset gate r_t to decide which information to pass forward the gate and which to exclude.

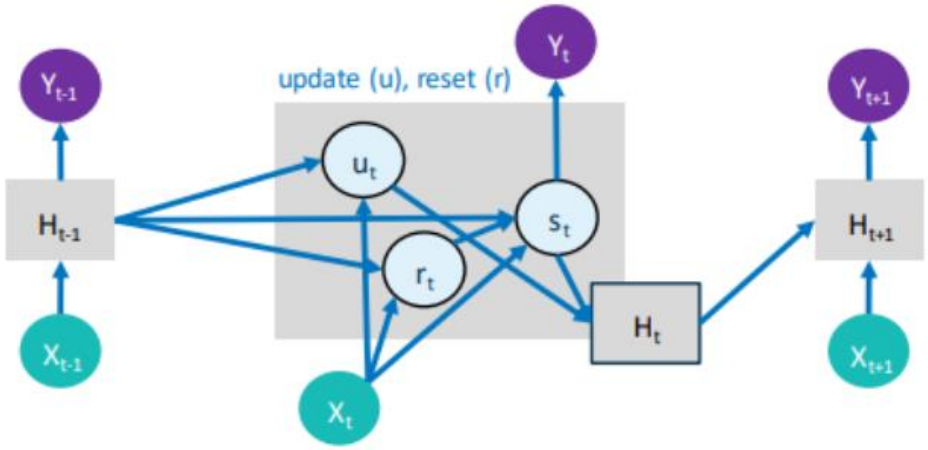

Figure 5.5: representation Of Gated Recurrent Unit Gate (SOURCE: SAS Institure)

The figure 5.5 represents the structure of a GRU unit cell, the Update u_t and reset r_t gates are computed by multiplying the previous hidden unit to some weight parameters then add the product of the input with more weight parameters.

The reset gate is used to calculate the intermediate state s_t; this gate receives the information from the inputs and the previous hidden states but multiplied by the reset gates. In this way, only some of the previous information reach this state.

The current hidden unit is calculated as sort of weighted average between the previous hidden unit H_{t-1} and the intermediate weight with weight the update gates. The concept is that the reset gate decides which information to discard and the update gates determine how much the intermediate states influence the hidden state. The output is computed by using the actual hidden state.

The following equations describe the Gated Recurrent Unit architecture.

$$u_t = sigmoid\ [W_u \cdot X_t + U_u \cdot\ H_{t-1} + b_u] \text{ (formula 5.5)}$$

$$r_t = sigmoid\ [W_r \cdot X_t + U_r \cdot\ H_{t-1} + b_r] \text{ (formula 5.6)}$$

$$s_t = tanh\ [U_s \cdot (r_t\ \cdot H_{t-1}) + W_s \cdot\ X_t + b_H] \text{ (formula 5.7)}$$

$$H_t = u_t \circ H_{t-1} + (1 - u_t) \circ s_t \text{ (formula 5.8)}$$

$$Y_t = sigmoid\ [W_Y \cdot H_t + b_Y] \text{ (formula 5.9)}$$

In practice:

- For r_t =0, the intermediate state contains only information about the current input.
- For u_t = 0, the intermediate state is used as current hidden state.
- For r_t = 0 and u_t = 1, the GRU architecture is identical to the RNN architecture
- For u_t = 2, the previous hidden state is used as current hidden state.

5.2.2 Long Short-Term Memory

The Long Short-Term Memory is a RNN architecture that has been proposed the first time by *Hockreiter and Schmidhuber in 1997*, subsequently a lot of modifications to the original LSTM have been implemented.

The main feature of the LSTM models is that it's able to carry information along long distance in time without requiring all the information flow through the network, selecting the information that are more useful to explain the series and in this way eliminating the problem of exploding or vanishing gradient.

Long-Short Term Memory has a network which is slightly different from the network of the GRU, in fact, the LSTM has four states: input i_t, forget f_t, a unit cell g_t, an output o_t and the intermediate state s_t as represented in the figure 5.6.

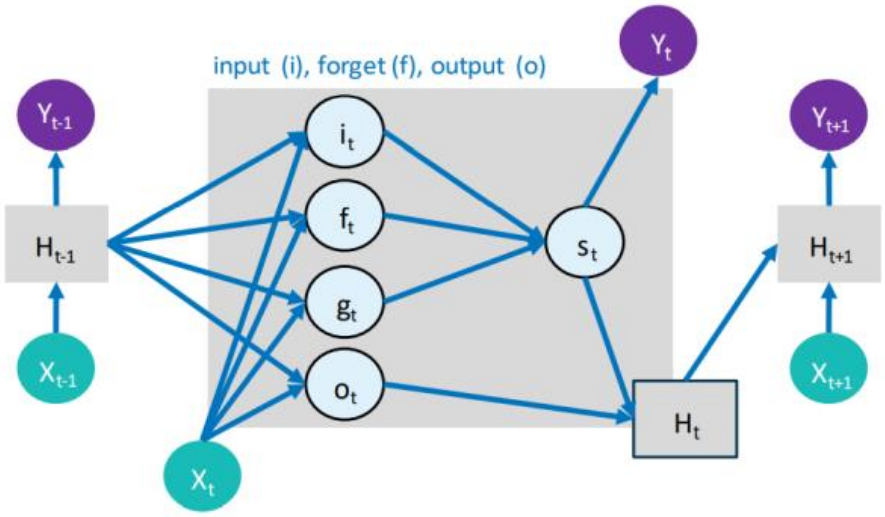

Figure 5.6: Long Short-Term Memory unit architecture (SOURCE: SAS Institute)

All the gates are obtained by multiplying weights by the previous hidden units and the current inputs and adding them together, so that all gates have the same structures, but they have different weights.

The input gate is used to determine how much details of the inputs have to be taken into account into the intermediate state.

The forget gate is used to understand how much of the previous intermediate state information have to be taken into account in the current intermediate state s_t.

The output gate decides how much information flows from the current hidden state to the output and the next hidden state.

The unit g_t combine the information from the input and the current hidden state, it can be considered as a sort of hidden state.

The intermediate state s_t is equivalent to a memory. It's a combination of previous information, coming from the input state, the forget state and the pseudo hidden state which is the unit g_t.

The equations that compose the architecture of an LSTM are the following:

$$i_t = sigmoid\ [W_i \cdot X_t + U_i \cdot\ H_{t-1} + b_i] \text{ (formula 5.10)}$$

$$f_t = sigmoid\ [W_f \cdot X_t + U_f \cdot\ H_{t-1} + b_f] \text{ (formula 5.11)}$$

$$o_t = sigmoid\ [W_o \cdot X_t + U_o \cdot\ H_{t-1} + b_o] \text{ (formula 5.12)}$$

$$g_t = \tanh\left[W_g \cdot H_{t-1} + U_g \cdot\ X_t + b_g\right] \text{ (formula 5.13)}$$

$$s_t = f_t \circ s_{t-1} + i_t \circ g_t \text{ (formula 5.14)}$$

$$H_t = \tanh[s_t] \circ o_t \text{ (formula 5.15)}$$

$$Y_t = sigmoid\ [W_Y \cdot H_t + b_Y] \text{ (formula 5.16)}$$

where W, U and b represent the different weights, the dot is the dot product of vector and the hollow dot is the element wise multiplication (Hadamard product).

In practice:

- For f_t = 0, the unit doesn't consider the information that comes from the previous state.
- For i_t = 0, the unit doesn't consider the new information

Both LSTM and GRU are trained with the same methodology typically used for training RNN.

5.3 Hyperparameters and Pre-processing

In a Neural Network model there are different forms of parameters: some parameters are calculated and trained on the model (i.e. model parameters), while others are not directly extrapolated by the model but they have to be chosen "a priori", they are called hyper-parameters.

The model parameters, in practice, are internal to the network, in the sense that they are learned automatically from the training sample, and they are used to make the predictions in a model (for example the weight of the neurons).

The hyperparameters are instead used to define the structure of the network or they are used to determine how the network is trained. For this reason, they should be set "a priori", in the way that they should be chosen before training phase.

There are different ways to choose the hyper-parameters, such as: Manual Search, Grid Search, Random Search, Bayesian Optimization.

In this master thesis some parameters are chosen by manual search, in other words, they are tuned by trial and error, by guessing the parameter and comparing the value of accuracy between different model with alternative parameters; other hyper-parameters are set through the dlTune action of SAS® Viya® software, as it will be explained in the chapter 5.3.9.

5.3.1 Hidden Layers and Hidden Neurons

The number of Hidden layers is one of the hyperparameters, in particular, it regards the "deep" in a deep neural network model. In fact, the number of hidden layers in a traditional Neural Network is the number of layers between the input and the output layer. This hyper-parameter depends on the model type, in general more hidden layers should improve the model until overfitting is reached.

In a Recurrent Neural Network model is not clear what the hidden layer is, some source claims that the model, when unfolded, has the same amounts of hidden layers as the number of time steps.

Taking in consideration that the hidden layers do not share the parameters, here we assume that the model specified in the figure 5.4 has only one hidden layer. So that in order to have more hidden layers, to the formulas regarding the figure, it can be stuck another layer.

The number of Hidden neurons or hidden units is an hyperparameters that regards the complexity of the model. Specifically, in increasing the number of hidden units the model should increase in fitting performance till it reaches the best performance and then it overfits. There are many ways to choose this hyperparameter, an interesting way is to plot the value of the error measure on the y axis and the number of neurons on the x axis and choose the point that minimize the test error measure.

5.3.2 Activation Function

The activation function, since it's defined at the model definition stage, can be considered as an hyperparameter.

There are different activations functions that are used in the general practice, it's also possible to define your own activation function. In the practice there some activation function which are commonly used: sigmoid function, hyperbolic tangent function (Tanh), Rectified Linear Unit (ReLu), Linear.

The activation function that have been mentioned can be defined in the functional formula as in the figure 5.7.

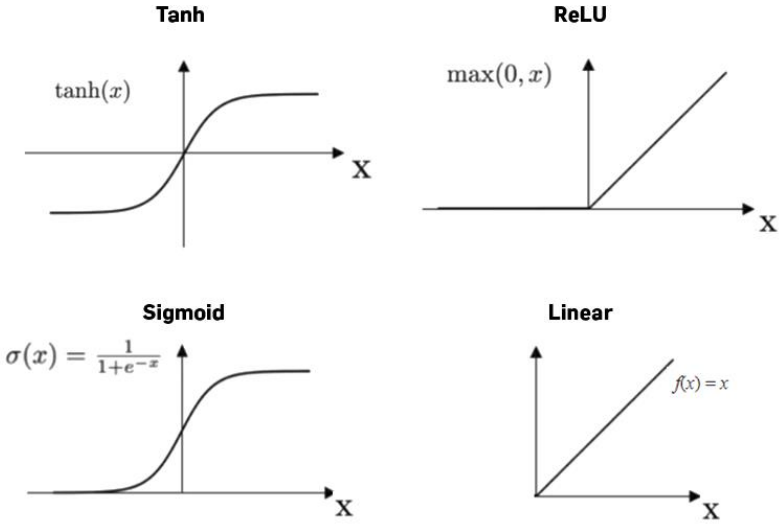

Figure 5.7: Tanh, Sigmoid, ReLU, Linear activation function (SOURCE: https://docs.paperspace.com/machine-learning/wiki/activation-function)

The use of the activation function depends on the target to study, for example, in the case of classification problem, a softmax activation function is used.

5.3.3 Number of epochs

The number of epochs represents the number of times the whole training dataset is passed forward and backpropagated through the model. The dataset is too big to be feed in a time, for this reason is useful to divide it in batch. In the end it is interesting to define the size of the batch that is a hyperparameter that characterize the number of samples that are passed into before the update of the model parameters.

5.3.4 Learning rate

The learning rate is an important hyperparameter. This number is usually in the interval between 0 and 1. When the learning rate is too high it can cause the overshooting of the point of minimum and for this reason it causes the divergence. In the opposite situation, it the learning rate is too low, it's not sure that the minimum is reached after the iteration that are set to compute the model.

In this thesis the learning rate is calculated with the SAS dlTune Action, after that an interval for this hyperparameter is set.

5.3.5 Test–Train ratio

Another hyperparameter that can be set is the training – test ratio. In machine learning and neural network modelling is useful to divide the dataset in train test and validation. Training data are that part of the dataset on which the model is trained, while validation data are the part that regards the validation of the model, and finally, the test set is the part that has been left out from the whole training phase in order to evaluate the goodness of the fit. In the case that there are too few observations to split the dataset in three part, it's possible to divide the data between training and test, that is the case of this series.

There are different proportion that are commonly used as test, train, validating split: train 80%, validating 10%, test 10% or in the case of only test train, often the split is 80% train and 20% test.

In general, the data are split between the different part by a random sample. Since in this thesis we are considering a time series, the data that are in a part or another cannot be randomly chosen but they should be in series as in the figure 4.3.

5.3.6 Preprocessing

Another important feature to be discussed regards the preprocessing of the data. In fact, in neural network modelling a thing that is supporting a stable convergence is scaling the data. There are different ways to scale the data, one that is very used is the standardization.

The standardization transforms the input data in a distribution of data which is rescaled and has mean equal to 0 and standard deviation equal to 1.

Herby there is the formula for the standardization that concern to subtract, to the observation, the mean of the distribution of the input and divide by the standard deviation.

$$x_{std} = \frac{x-\mu}{\sigma} \text{ (formula 5.17)}$$

where μ represents the mean of the distribution and σ is the standard deviation of the distribution.

5.3.7 Loss function

The objective function is another hyperparameter. This function is used to understand how well the model fits the data, in particular, it's maximized or minimized in order to optimize the value of the parameters. Defining L the loss function and θ the set of parameters and biases of the model, the aim is to find the weights that satisfy the following equation:

$$\arg min_{\theta}\big(L(\theta)\big) \text{ (formula 5.18)}$$

When facing a regression problem as the time series is, there are at least three different objective function: Mean Squared Error (MSE), Mean Absolute Error (MAE), Sum of Squared Error (SSE). In this thesis the loss function that is used to optimize the structure is the MSE, as it's defined in the formula 5.18.

5.3.8 Optimizer

There is a path to follow in order to calculate the parameters which identify the neural network model. First of all, the input data are feed into the network, then they pass through the neurons and the different hidden layers, in this way, the value of the output is computed. Then the loss function defined in the formula 5.18 is calculated by comparing the real output with the computed one and optimized. Then, in the third step of the backpropagation algorithm is the total loss is used, in fact, starting from the output and going back to the input layer, the gradients of the defined objective function are computed with respect to the weight and the biases. Finally, the values of the computed parameters are optimized in order to minimize the total error.

In the context of Feed Forward Neural Network, we have that only the input goes into the hidden layer, while in the case of Recurrent Neural Network, we have that jointly to

the input, also the hidden states goes into the hidden layer, in this way there is a sort of parameter sharing, as explained in chapter 5.2.

In order to update in the best way the parameters, an optimization algorithm is used. In general, the most common algorithm is the Gradient Descent. In the paper "An overview of gradient descent optimization algorithms" written by Sebastian Ruder, the algorithm is defined as: "Gradient descent is a way to minimize objective function L(θ) by updating the parameters in the opposite direction of the gradient of the objective function $\nabla_\theta L(\theta)$ with respect to the parameters. The learning rate η determines the size of the steps we take to reach a (local) minimum".

In this way the Gradient Descent feeds the whole training dataset, calculates the whole error and then updates the parameters.

The update is represented as:

$$\Delta\theta = \theta - \eta \cdot \nabla_\theta L(\theta) \text{ (formula 5.19)}$$

where $L(\theta)$=$\sum_{i=1}^{N} L_i(\theta)$ with N the number of training examples.

The problem with this optimizer regards the fact that it's slow and intractable.

A first extension of this optimizer is the Stochastic Gradient Descent. This optimizer takes randomly training examples and then make and update of the parameters at each iteration. The representation of this process is the following:

$$\Delta\theta = \theta - \eta \cdot \nabla_\theta L_i(\theta) \text{ (formula 5.20)}$$

where i is the i-th example.

Adopting this methodology lead to a faster convergence, in particular, when dealing with big dataset, but a negative result is that the stability of the optimization with respect to different iteration decrease.

In dealing with time series and LSTM or GRU usually it's good to use the ADAM optimizer.

ADAM (Adaptive Movement Estimation) is an optimizer that is much more complex but can increase the performance. In practice ADAM is a method that computes adaptive learning rates for each parameter, it has a different learning rate for each parameter and performs an update, to the features that appear more or less frequently. In addition, it prevents a decay of the learning rate which could cause the model to stop learning.

The equations that compose this optimizer for only one iteration are:

$$g_t = \nabla_\theta L_t(\theta_{t-1})$$ (formula 5.21)

$$m_t = \beta_1 \cdot m_{t-1} + (1 - \beta_1) \cdot g_t$$ (formula 5.22)

$$v_t = \beta_2 \cdot v_{t-1} + (1 - \beta_2) \cdot g_t^2$$ (formula 5.23)

$$\widehat{m}_t = \frac{m_t}{(1-\beta_1^t)}$$ (formula 5.24)

$$\hat{v}_t = \frac{v_t}{(1-\beta_2^t)}$$ (formula 5.25)

$$\theta_t = \theta_{t-1} - \eta \cdot \frac{\widehat{m}_t}{\sqrt{\hat{v}_t}+\epsilon}$$ (formula 5.26)

where m_t is the mean and v_t is the variance of the gradients g_t: these values are used to update the parameters. The variables $\beta_1$and β_2 are the value that regards the decay of the moments previously defined, their values are contained in the interval 0, 1 with 1 excluded. The g_t^2 represents the element-wise square $g_t \circ g_t$ and all operations on vectors are element-wise. Finally, the variable ϵ is used to make that the denominator is different from 0. The last equation finally represents how the parameters are updated. This cycle has to be done until θ_t is not converged.

In general, with respect to other methods, it converges very fast and it works generally good with neural networks.

5.3.9 Hyperparameters tuning

In order to select some hyperparameters, it can be used the hyperband method (*Li, DeSalvo, Rostamizadeh and Talwalkar 2017*), this methodology is interesting since the resources are adaptively allocated to the model structure that are promising.

The Hyperband method is implemented in the dlTune Action of the software SAS Viya.

This methodology starts by selecting a set of hyperparameter using a Latin hypercube sample, this sampling methodology is more accurate and efficient with respect to a random sample, which is the basic one. In fact, what can be obtained is a uniform sampling space, and if the hyperparameters are evaluated on a hold-out sample, it prevents overfitting.

The first step the set of sampled hyperparameters is assigned to the specific model structure, and the resources are allocated proportionally through via the hyperband method. Then the models are trained for one or more epochs and the less effective models are removed from the training process. Then the resources are reallocated and

distributed among the remaining models, and then these models are trained for one or more epochs and their accuracy evaluated. Finally, the process is repeated until a set of hyperparameters remains.

There are a lot of hyperparameters that can be tuned using this procedure, when dealing with ADAM optimizer it could be interesting to tune the learning rate, the β_1 and the β_2.

5.3.10 Deep Learning Action Set SAS®

In order to build the Neural Networks models it's possible to use SAS® Viya® Deep Learning CAS Action.

SAS® Viya® is a cloud-enabled, in-memory analytics engine that has a series of tools to produce quick, accurate and reliable analytical insights.

In this framework is possible to find the CAS Actions, that combined together are able to perform complex tasks. As defined in the SAS documentation: "CAS actions are organized with other actions in an action set and usually contain actions that are based on common functionality". The language that is used in the CAS actions is slightly different from the traditional SAS language, it's called CASL.

In particular, the Deep Learning Action Set provides a superset of actions for modeling and scoring with Deep Neural (DNN), Convolutional (CNN), and Recurrent (RNN) networks.

There available actions and their use, as found on the SAS documentation is the following:

Action Name	Description
addLayer	Adds a layer to a Deep Learning model
buildModel	Creates an empty Deep Learning model
dlExportModel	Exports a Deep Learning model
dlImportModelWeights	Imports model weights from an external source
dlJoin	Joins the data table and the annotation table
dlLabelTarget	Assigns the target label information

dlPrune	Prunes a layer in a Deep Learning model
dlScore	Scores a table using a Deep Learning model
dlTrain	Trains a Deep Learning model
dlTune	Tunes hyperparameters for Deep Learning model
modelInfo	Displays model information
removeLayer	Removes a layer from a Deep Learning model

(SOURCE: Deep Learning Action Set SAS® Documentation)

In this master thesis the Deep Learning models have been computed some of these Actions.

Chapter 6

Empirical application on different countries

In this chapter it is presented a comparison of an application of the Recurrent Neural Network with Long Short-Term Memory unit (LSTM) and Gated Recurrent Unit (GRU) structures to the classical Lee-Carter model and an ARIMA model to the classical Lee-Carter model.

The aim is to predict the future mortality showing how the classical structure of the Lee-Carter model can be improved in the forecasting ability by using the LSTM and GRU architecture by comparing their performances to the performance of an ARIMA model.

Finally, it's provided a comparison between the pricing of two kind of annuities in the case of using a LSTM, a GRU and an ARIMA, to project the forecast in the future. In addition to this, it's compared the previously identified structure with a benchmark structure based on the real probabilities that are taken from the Human Mortality database on another kind of annuity.

For these purposes, it has been developed a set of experiments on different countries used to compare the ability of the different models, in order to understand if the classical model can be improved using a RNN based model.

The countries that have been taken into account are Australia, Italy, Japan and USA, only male population has been considered.

6.1 Introduction to the experiments

The classical Lee-Carter model, as explained in the chapter 4, adopts a singular value decomposition to extract three parameters that combined can be used to estimate the mortality.

In particular, the three parameters are: α_x , β_x , k_t. The method to obtain the forecast of the mortality, developed by Lee and Carter in 1992, concerns the projection of the k_t parameter on a future time t. In fact, as can been denoted, this parameter is time dependent, differently from α_x , β_x that are function of the age of the person.

In order to fit the classical Lee-Carter model, it has been used the *lca* function from the package Demography.

The model has been fit on each country for the period represented in the table 6.1, while the age x, the subscript x of the parameters α_x , β_x, has been considered in the interval 0 to 100.

COUNTRY	PERIOD
Australia	1921 – 2018
Italy	1920 – 2017
USA	1933 – 2018
Japan	1947 – 2018

Table 6.1: Lee-Carter model fit period

This simplification has been used in order to reduce the volatility that is usually present in the extreme ages. This fact leads to have the table that is not closed, since we have that the probabilities are estimated only until the age of 100 years. For this reason, a hypothesis on the behavior of the mortality after age x=100 has been introduced in order to price the contracts that will be taken into consideration.

The classical Lee-Carter model adopts a random walk with drift to perform the projection of the parameter k_t, while in this thesis, as it has been done in the article "*A Deep*

Learning Integrated Lee-Carter Model" by *Negri et al*. we will consider the best ARIMA to project this parameter.

The comparison is made between the performance of the projected k_t of an ARIMA, a LSTM and a GRU architecture.

Since we are dealing with Neural Network modelling, there is the need to define a training-test split. This subdivision is used in order to train the model on training data and evaluate the performance of the model on the test data. This approach is applied both on the ARIMA, estimating the model on the training data and on the LSTM and the GRU.

The test-train split used is the following: nearly 80% of the time series data are used to train the model and 20% of the data are utilized to test the model. This approach has been performed to each country and the table 6.2 is summing up the periods.

COUNTRY	TRAIN	TEST
Australia	1921-1999	2000 – 2018
Italy	1920-1999	2000 – 2017
USA	1933-1999	2000 – 2018
Japan	1947-1999	2000 – 2018

Table 6.2: Train – Test split for each country

Different error measures can be applied to evaluate the performances of the models, usually the most used are Mean Absolute Error and Root Mean Squared Error, in fact, these two measures, in addition to the graphical check, are utilized to evaluate the forecast ability of the models. In other words, the best model is the one that obtains the minimum error.

The datasets that are used are downloaded from the Human Mortality Database (*www.mortality.org*). This website collects detailed mortality and population data that are used by researchers, students, journalists, policy analysts to perform longevity analysis.

The time series of the k_t that is the outcome of the Lee-Carter Model application is represented in the figure 6.1.

The vertical line in each graph represents the split between train and test series. In fact, after the vertical line, in each plot, the series is the test partition, while before there is the train partition.

All the time series have a non-linear decreasing trend, some of them decrease with a higher speed, while others much slowly. The decrease means that the mortality is improving. The volatility seems to be higher in the Italian series and, in particular, it has a peak in correspondence of the years of the second world war, since in that circumstance the mortality has increased a lot.

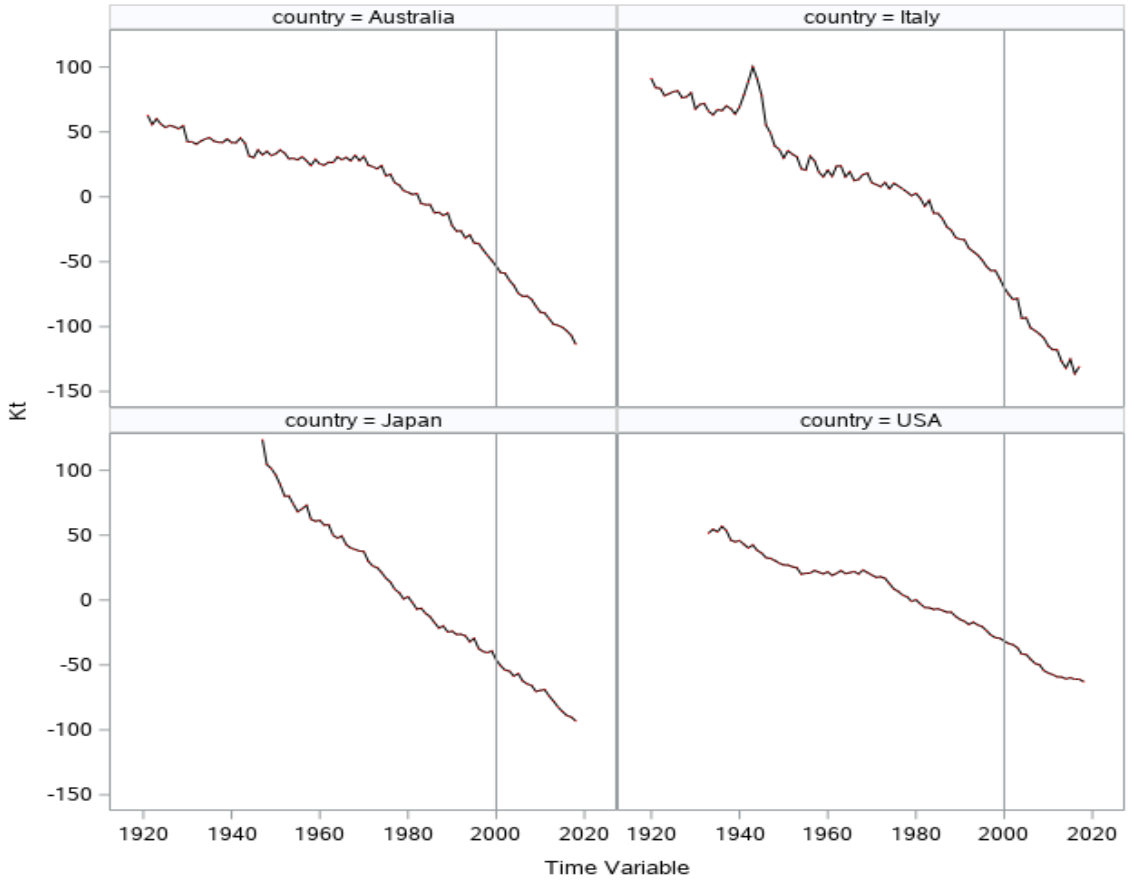

Figure 6.1: k_t time series in each considered country

The time series is divided in two parts, as explained before: training data and testing data.

The aim is to train the models on the train data and then compare the forecast with the test set, which has been left out from the training session, by calculating an error indicator such as the Root Mean Squared Error or the Mean Absolute Error.

The next figure is representing the Autocorrelation Function and the Partial Autocorrelation Function in the case of the whole series of k_t in the case of Australia.

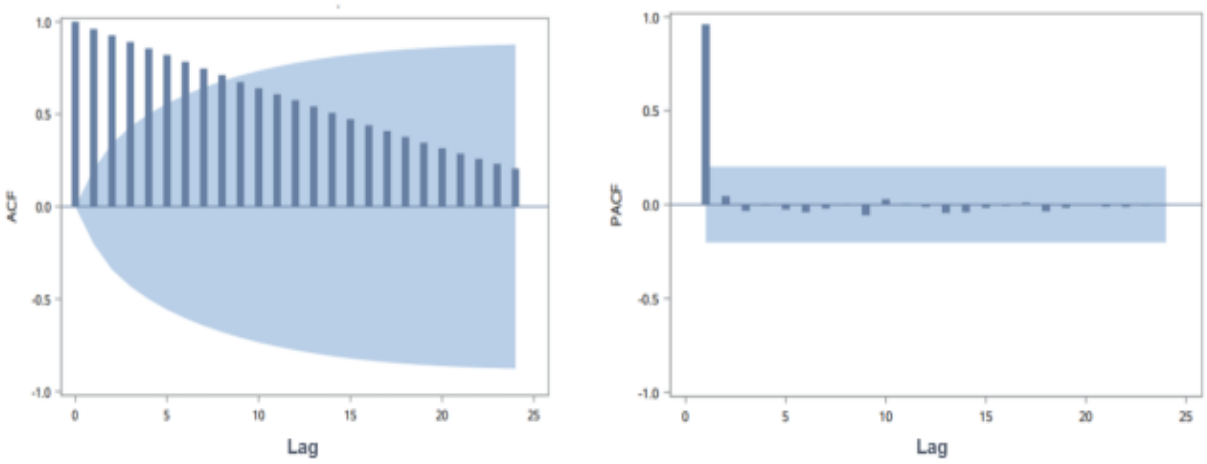

Figure 6.2: ACF and PACF of Australian k_t time series

From the ACF in the figure 6.2 it is clear that there is a decreasing trend, this support the hypothesis of non-stationarity of the series, so that a transformation is needed, in addition looking at the PACF is interesting to notice that the lag of value one of k_t, in other word k_{t-1} is significant in explaining the value of k_t . This explanation supports the hypothesis that there is correlation in the time series, and that it's possible to create a model to exploit this correlation.

The same thing can be done with all the remaining time series as a first exploration that prepare the ARIMA modelling.

6.2 Arima Models

In this chapter there is the explanation of the application of an ARIMA model to the k_t parameter of the Lee-Carter.

In contrast to the Classical Lee-Carter model on the paper developed in 1992, this experiment utilizes an ARIMA model.

In order to choose the ARIMA model and in particular the parameter p, d, q of this method, the *auto.arima* function available in the R package forecast, that is explained in the chapter 4.2.1.

The best model is computed on the training data as represented in table 6.2 for each country.

The best models according to the AIC are resumed in the table 6.3.

COUNTRY	BEST ARIMA (p,d,q)	AIC
Australia	ARIMA (1,1,0) with drift	421.89
Italy	ARIMA (0,1,0) with drift	502.45
USA	ARIMA (0,1,0) with drift	288.21
Japan	ARIMA (2,2,2)	281.41

Table 6.3: Best Arima and AIC measure in each country

Comparing the AIC in the different countries, it seems that there is a relation between the number of observations in the training set and the value of the AIC. In fact, the AIC is lower where the number of observations is lower, this is by construction; for this reason, it's not fully reliable to make direct comparison between countries.

In this case, this methodology is often in accordance to the classical Lee-Carter, in facts, for what concerns the Italian and the American time series the model that is proposed is a random walk with drift.

The statistics regarding the Box-Pierce test indicate that in all models the residuals are uncorrelated, since the p-value lead to accept the null hypothesis.

In addition to this, as it can be seen in the plot, residuals are normally distributed with mean approximatively equal to 0.

After that the model is trained, the performances are calculated on the test set, as defined in the table 5.2. The error measure that it has been used and calculated is the Root Mean Squared Error and the Mean Absolute Error.

COUNTRY	MAE	RMSE
Australia	21.08	23.19
Italy	24.54	26.76
USA	8.62	9.64
Japan	11.52	13.03

Table 6.4: ARIMA: RMSE and MAE in each country

The RMSE and the MAE are the lowest in the case of the American series, but in general it's not possible to make comparison between the different countries, since this measure is a scale dependent measure.

It could be useful to make a comparison between the rank of the RMSE with respect to the AIC, the ranking is conserved for all, except the USA and the Japan, which have opposite positions. In fact, USA has lower RMSE comparing it with the Japan; this could mean that the ARIMA is a good model to project the series, or in contrast, the ARIMA is not the best model to forecast the time series.

In the next chapters it will be possible to make the comparison between the ARIMA and the Neural Networks models.

6.3 Neural Networks Modelling

In this chapter there is presented the application to the time series of the parameter k_t of the Long-Short Term Memory and the Gated Recurrent Unit architecture.

These models are the best Neural Network model to be used since, as explained in the chapter 5, they are able to capture the pattern inside the time series data.

The approach that it has been used follows the notations used in the paper "A Deep Learning Integrated Lee–Carter Model" written by Andrea Nigri et al.

The aim is to build a LSTM and GRU model that is able to fit and approximate the function f witch links k_t to its own lag.

The relation can be presented as follows:

$$k_t = f(k_{t-1}, k_{t-2}, \dots, k_{t-j}) + \varepsilon_t$$

Where $j \in \mathbb{N}$ is the number of lag considered and ε_t the homoscedastic error component.

This application resides under the supervised learning problem, since a target variable is present and the input variable are used to explain and predict the target.

The input-output table can be built in this way:

Output		**Input**		
k_t	k_{t-1}	k_{t-2}	…	k_{t-j}
k_{t+1}	k_t	k_{t-1}	…	k_{t-j+1}
k_{t+2}	k_{t+1}	k_t	…	k_{t-j+2}
…	…	…	…	…
k_{t+n}	k_{t+n-1}	k_{t+n-2}	…	k_{t-j+n}

Table 6.5: Dataset for the supervised learning

The aim is that after that the input-output functional relation f is estimated the input, which is composed by the lagged value of k_t is able to predict the output.

In practice the input is a (n x j) matrix of the time lags k_t and the output is the (n x 1) vector of its current values.

In order to obtain the forecast the values of k_t at time n+1, n+2, …, n+j are obtained in a recursive manner. More in general, the values of the forecasted variable $\hat{k}_t$ in a generic time n+τ is fitted using the values of the k_t, with t in the list n+τ-1 , n+τ-2, …, n+τ-j as unique input.

An important thing to notice is that going forward into the time, the forecasted values of $\hat{k}_t$ are calculated recursively using only the forecasted value $\hat{k}_t$ that are not yet observed and for this reason there is no way to calculate the forecast accuracy above these values.

The first step of this analysis as in the case of the ARIMA is to fit the Lee-Carter model and estimate the parameters α_x , β_x , k_t through the Singular Value Decomposition.

Then taking into account only the time series of k_t, we apply the training test split as depicted in the previous chapter.

Considering the paper "A Deep Learning Integrated Lee–Carter Model" written by Andrea Nigri et al. the analysis has been performed by considering only one time lag, in other words j=1.

The software that have been used is SAS Viya, as described in the chapter 5.3.10, in particular, the Deep Learning Action set has been used.

In order to find the best Neural Network model, a first fine tuning has been carried out both for LSTM and GRU models in order to select some hyperparameters. This step has been made only on the Australian k_t time series, and then applied in the same way on the other countries. The approach used in this phase is trial and error approach, for this reason the details are not explained in their particular in this instance.

This first best combination of hyperparameters is used to tune the model in a more accurate way.

The results of the first tuning session has been useful to identify the architecture of both the Neural Network model, in fact, a structure with a single hidden layer is the best structure in both cases.

For what concern the activation function, there were no evidence to insert an activation function which was different than a linear one. The Relu function is working in a good way too, with respect to the others activation functions. In the end the choice was to keep the most simple and classical structure using a Linear activation function.

The following hyperparameters are then the same for Long Short-Term Memory and Gated Recurrent Unit models, since the two models behave in a similar way.

The number of neurons has been chosen looking at the behavior of the RMSE with respect to the neurons.

For what concern the preprocessing, the input has been standardized to have a distribution with mean 0 and standard deviation equal to 1.

The initialization of the random weight as been fixed to normal, so that the weight has mean 0 and standard deviation 1 at the beginning.

The loss function that has been chosen is the Mean Squared Error as it is usual in training these kinds of models.

For what concern the optimizer, the ADAM optimizer has been used for both models, since many papers regarding these kind of models uses it and it seemed to get the best performance in term of stability.

The models after that their structure has been defined, it has been tuned using the DlTune CAS Action, that been explained in the chapter 5.3.10, this algorithm requires additional hyperparameters to be set, some of them are filled as intervals, other in a punctual way and then the algorithm find the best model according to the hyperparameter chosen.

The minibatch size has been set equal to 20, the number of maximum epochs is equal to 100, the tune iteration is set to 75 and the number of trials 50. For what concern the hyperparameters that have been set as intervals in which the algorithm automatically find the best, the learning rate is between a lower bound=0.00001 and an upper bound=0.1 and both β_1 and β_2 are between a lower bound=0. 1 and an upper bound=0.999. The learning policy is made at step, the maximum gradient clipping equal to 1000 and the minimum gradient clipping equal to -1000.

The hyperparameters are chosen on the validation table that has been set equal to the test dataset.

The next step regards the choice of the last hyperparameter which is the number of neurons.

The number of hidden units in the hidden layer has been chosen comparing the value of the Root Mean Squared Error, that it has been found comparing the forecasted value of k_t with respect to the real value of k_t in the test dataset.

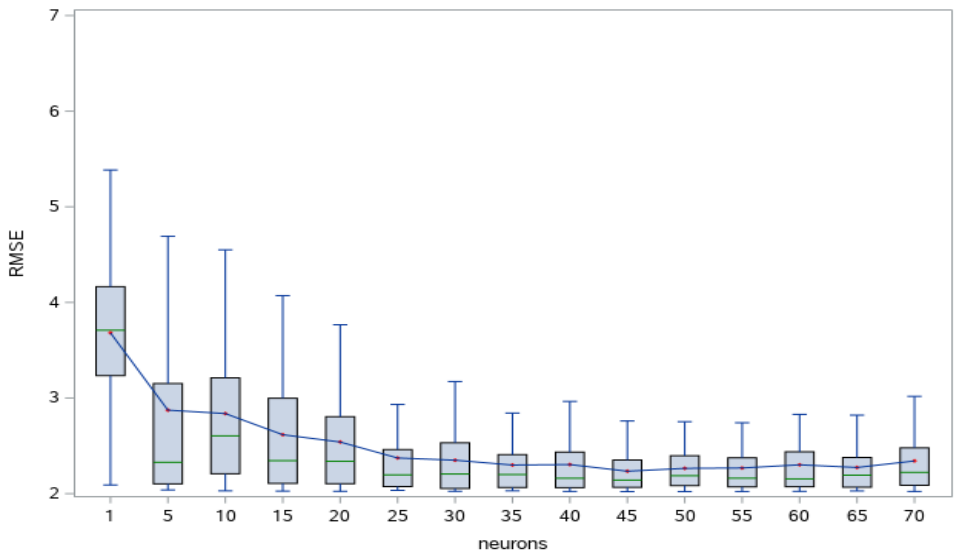

Figure 6.4: RMSE by neurons in the case of Australia and Gated Recurrent unit model

The figure 6.4 shows the behavior of the Root Mean Squared Error with respect to the neurons. The plot has been created by training 100 models with the same structure, number of hidden neurons and hyperparameters, changing each time the seed, defined the neurons.

The plot can help to choose the number of neurons, in fact the best model is the one witch stability is higher and witch Root Mean Squared Error is lower in mean.

In general the mean of the RMSE should have a minimum and then increase, the increasing behavior means that the model is overfitting, and this happens usually when the number of neurons increase a lot, while the opposite, when there are too few neurons, it should underfit, and also in this case the RMSE should be high.

In this case the minimum is reached at 45 neurons, and then the RMSE start increase slowly. In fact, following this approach the number of hidden units has been chosen equal to 45 in the case of the Gated Recurrent Unit Model.

The same procedure has been applied to all the countries taken into consideration for both GRU and LSTM structure.

The results are showed in the table 6.6.

COUNTRY	Neurons for the LSTM	Neurons for the GRU
Australia	8	45
Italy	17	37
USA	27	46
Japan	15	41

Table 6.6: Number of neurons in each country for LSTM and GRU structure

From the table 6.6 it's possible to notice that the number of neurons chosen in the case of the GRU structure is higher, in mean is 42, with the number used for the LSTM structure, which is in mean 17. This fact can be associated with the fact that the GRU is a simpler model. In fact, the number of parameters associated with the GRU unit is lower

with respect to the LSTM unit, and this causes overfitting at a number of neurons, weight, which is higher. In addition, it is interesting to observe that the volatility associated with the number of neurons is higher in the case of the LSTM.

After that the number of hidden neurons has been set, the model can be trained using the DlTune Action and it's possible to obtain the forecast using the DlScore CAS Action.

To sum up, the model has been trained until this point on the training data, as defined in the table 6.2, and then the forecast has been evaluated on the test data.

The error measure that has been used is the Root Mean Squared Error and Mean Absolute Error, as in the case of the ARIMA model in the previous chapter.

The results are presented in the following tables:

COUNTRY	GRU - MAE	LSTM - MAE	GRU - RMSE	LSTM - RMSE
Australia	1.84	2.35	2.26	2.82
Italy	3.26	3.84	4.30	4.27
USA	2.75	2.32	3.28	2.91
Japan	5.54	4.51	6.69	5.30

Table 6.7: GRU, LSTM: RMSE and MAE in each country

The results in the table 6.4 show that the best model, according to RMSE is the GRU in the case of the Australian time series. In contrast the worst model is the one of the Japanese time series.

This comparison is not strictly significant since both RMSE and MAE are scale dependent, anyway it gives an idea. It's more useful to present the comparison between models in the same country.

In not all the cases a lower RMSE in a model corresponds to a lower MAE for the same model compared to the others. This is the case of the Italian country, where the minimum MAE is owned by the GRU architecture, while the minimum RMSE is the one of the LSTM. Looking at the magnitude of the numbers it's then easy to see that the best model should be the GRU, since the MAE is lower in the case of the last model, while the RMSE are approximatively identical.

Finally, in general, it has been observed that the volatility related to the RMSE in changing in the seed is in mean lower in the case of the GRU model, suggesting that this kind of model is more stable with respect to the LSTM.

6.4 Models comparison and calculation of annuities

In this chapter the models that have been computed in the previous chapters are compared under different point of view.

A first comparison regards the forecast of the k_t parameter, then the probabilities that are computed using this parameter through the Lee-Carter model and finally the calculation of different kind of annuities is obtained from the forecasted probabilities and compared.

In the following plot there is the forecast of the different models that have been taken into consideration (ARIMA, GRU, LSTM) for each country taken into consideration.

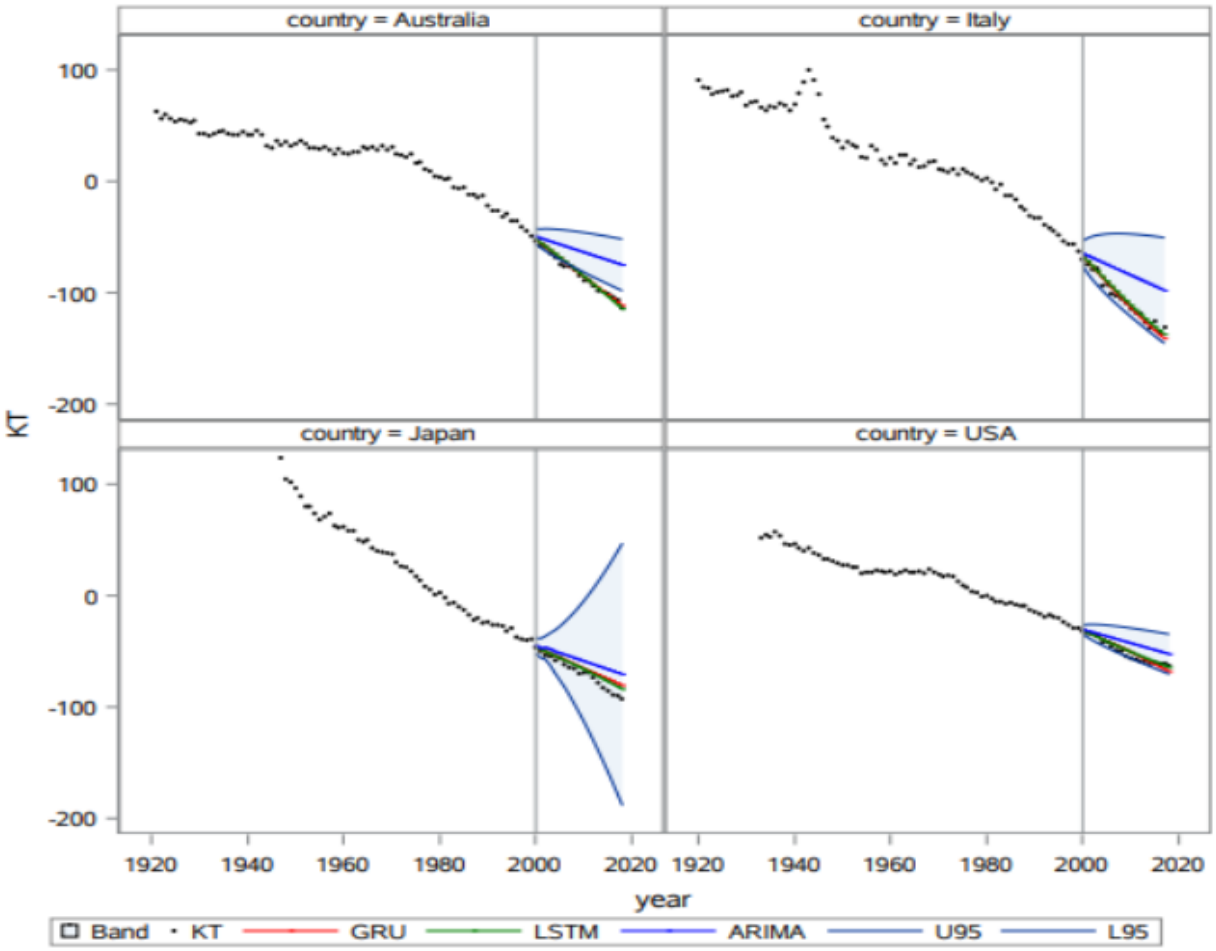

Figure 6.5: Forecasted time series in the test set for different models and countries

As it can be easily seen from the forecast the GRU and the LSTM models are predicting in a more accurate way the future k_t.

Looking at the figure 6.5, it's not so clear where the GRU is predicting better than the LSTM, indeed, the two models are quite similar in their predictions.

COUNTRY	ARIMA	GRU	LSTM
Australia	23.19	2.26	2.82
Italy	26.76	4.30*	4.27*
USA	9.64	3.28	2.91
Japan	13.03	6.69	5.30

Table 6.8: Comparison of the RMSE per model and country

The conclusions that come out from the graphical representation are founded. In fact, looking at the RMSE the ARIMA model has always the worst performance, while the GRU and LSTM models have results that are comparable, and similar.

Considering both MAE and RMSE, the GRU architecture performs better on the test data on three of the four countries taken into consideration, even if some values of the RMSE are very similar. In fact, where the values are signed with an asterisk it means that in that case the MAE is not having the same behavior that has the RMSE.

The next step is the comparison of the probabilities of death within each country for each model that it has been computed in the previous chapters.

In order to calculate these probabilities, the coefficients α_x and β_x have been calculated using the Lee-Carter SVD for an interval of years that is equal to the whole time series considered when calculating the factor k_t, and the interval of ages between 0 and 100.

The results in terms of deaths probabilities are represented in the next figure.

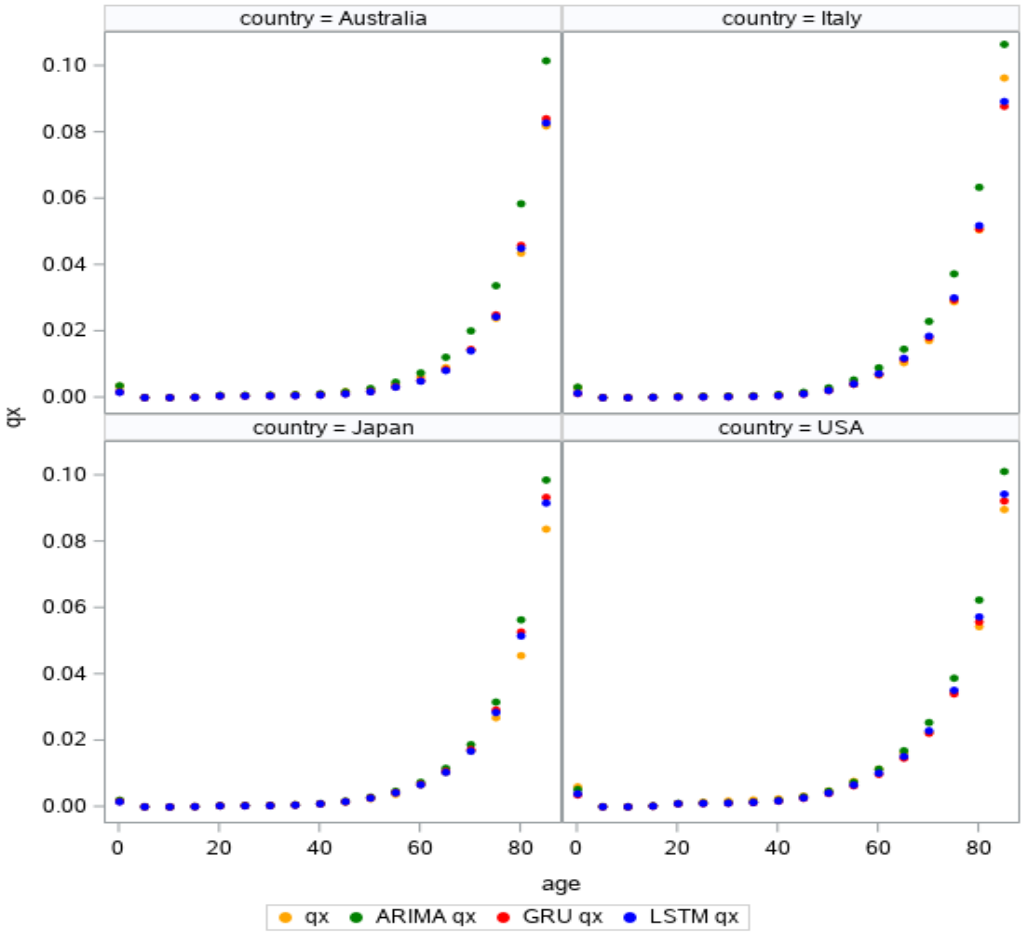

Figure 6.6: Death probabilities in each country for the different models

In the figure 6.6 the death probabilities are represented in the year 2017 for the Italian population and the year 2018 for the others country for the different models analyzed (Arima, GRU, LSTM) compared with the real probabilities that has been taken from the Human Mortality Database.

What is clear from these graphs is that in general, the models have a good accuracy within the year interval 5 – 70 years.

There are some differences in the probability to die within the first years of life, in fact, in the American population the probabilities are underestimated by the model, while in the case of the Italian and the Australian population, these are underestimated by the LSTM and the GRU but are correctly estimated by the ARIMA model. in the case of the Japan population, all the models correctly estimate the mortality.

Concerning deaths probabilities for age higher than 70 years, in general, all the models overestimates the mortality. There is an exception for the Italian case where the mortality is underestimated in the LSTM and the GRU models.

In the end, it's clear that the LSTM and the GRU models outperform the ARIMA models in the death probability considered.

Since there is an error that is in common to all the models, it means that this error comes from the use of the Lee-Carter model, in other words, the factor α_x and β_x.

In the last years a lot of extension to the classical Lee-Carter have been proposed. One of the major deficiencies of the Lee-Carter model is the cohort effect is not taken into account. In fact, in this model, the particular cohort in which the person has born is not accounted as a factor that is useful to explain the future and actual behavior of the mortality.

Another interesting curve is represented in the next plot.

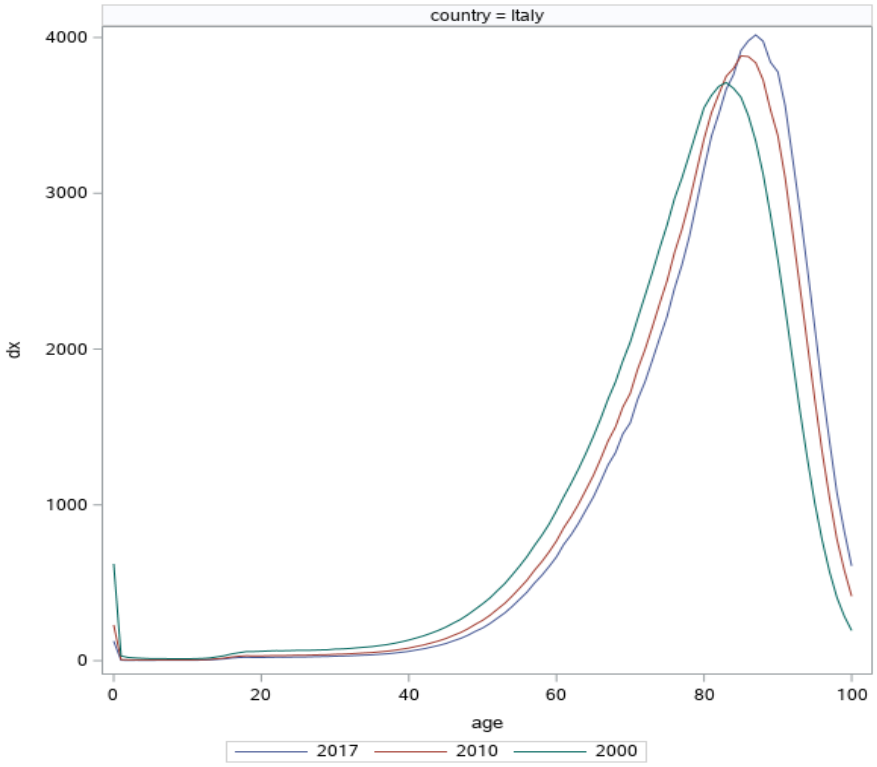

Figure 6.7: Curve of deaths for the Italian population at different years

The figure 6.7 represents the curve of deaths estimated using a GRU model at the different age, in the case of the Italian population for different years.

What can be clearly seen is that the curves have the behavior of the rectangularization and expansion that has been explained in the chapter 2. In facts, as it's possible to see, the behavior of the curve is clear. In increasing in the time, going from 2000 to 2017, the deaths during the first age of life are decreasing, people are surviving longer and at older ages. This trend is confirmed by the peaks of deaths that is higher in 2017 and is shifted.

In the next lines it will be presented a calculation of three different kind of annuities for the models that has been considered. This experiment has been made in order to see the differences in term of pricing of annuities of the different models.

The different contracts have been identified in the next table, and denoted with the letter A, B, C for simplicity.

Annuity	Gender	Age	Deferral period	Temporary	Payment	Rate	Year subscription
A	m	25	42	ω-67-1	Advance	0.01	2000
B	m	67	0	ω-67-1	Advance	0.01	2000
C	m	67	0	18 or 17	Arrears	0.01	2000

Table 6.9: Description of the different kind of contracts taken into consideration

The formula for the calculation of the unitary annuity of kind A is the following one:

$$ {}_{m/}\ddot{a}_{x} = \sum_{h=m}^{\omega-(x+m)-1}(1+i)^{-h}\,{}_{h}p_{x} \text{ (formula 6.1)} $$

where m is the deferral period, x is the age of the individual, and ${}_{h}p_{x}$ the probabilities are calculating through the cohort life table for the individual aged x in the year of the subscription.

The formula for the calculation of the unitary annuity of kind B is the following one:

$$ \ddot{a}_{x} = \sum_{h=0}^{\omega-x-1}(1+i)^{-h}\,{}_{h}p_{x} \text{ (formula 6.2)} $$

where x is the age of the individual, and ${}_{h}p_{x}$ the probabilities are calculating through the cohort life table for the individual aged x in the year of the subscription.

The formula for the calculation of the unitary annuity of kind C is the following one:

$$ a_{\overline{x:n}|} = \sum_{h=1}^{n}(1+i)^{-h}\,{}_{h}p_{x} \text{ (formula 6.3)} $$

where x is the age of the individual, and ${}_{h}p_{x}$ the probabilities are calculating through the cohort life table for the individual aged x in the year of the subscription and n is the number of annual benefit receipt.

Since the Lee-Carter model has been calculated using and interval of age between 0 and 100, the last probability to die is not equal to 1. In order to close the table, in fact, the method that it has been used is to project linearly the curve of the obtained probabilities of deaths of the last five ages (95-100), until the probability of death equal to 1 is reached.

All the policies start in the year 2000, that in practice should be observed, in fact, this is a simplification, that is used only for the scope to compare the different models to a benchmark model that it has been computed using the real probabilities, that have been taken from the HMD.

The policy of kind C has different temporary duration, in facts, the Italian duration is of 17 years (i.e. until 2017), while the others countries have a duration of 18 years, this is due to the fact that the last year at disposal for the Italian death probabilities is the 2017.

Model	Type A	Type B	Type C	RMSE on k_t	MAE on k_t
ARIMA	10.06	15.06	12.58	23.19	21.09
LSTM	12.77	15.91	12.99	2.83	2.36
GRU	11.88	15.88	13.00	2.27	1.85
BENCHMARK	/	/	13.20	/	/

Table 6.10: Annuities results on the Australia

Model	Type A	Type B	Type C	RMSE on k_t	MAE on k_t
ARIMA	9,84	14,70	12,01	26.76	24.55
LSTM	10,32	15,49	12,33	4.27	3.84
GRU	11,25	15,57	12,36	4.30	3.26
BENCHMARK	/	/	12,48	/	/

Table 6.11: Annuities results on the Italy

Model	Type A	Type B	Type C	RMSE on k_t	MAE on k_t
ARIMA	9,05	14,51	12,10	9,65	8.63
LSTM	9,77	14,83	12,27	2,91	2.32
GRU	10,78	14,87	12,27	3,29	2.75
BENCHMARK	/	/	12,48	/	/

Table 6.12: Annuities results on the USA

Model	Type A	Type B	Type C	RMSE on k_t	MAE on k_t
ARIMA	10,44	15,48	12,84	13,04	11.52
LSTM	12,35	15,82	12,99	5,31	4.51
GRU	11,15	15,76	12,97	6,69	5.54
BENCHMARK	/	/	13,05	/	/

Table 6.13: Annuities results on the Japan

As expected, the price of the annuities is different basing on the type. In fact, type B annuities have the highest price. The reason is that the probabilities to survive and get the annual amount are higher with respect to same probabilities in the case A and with respect to the case C, the probabilities are the same but the number of future amount to be given to the policyholder is lower.

The price of the annuities that is higher for each type is the type B, since the probabilities to take advantage of getting a higher number of annual benefits is bigger than for the type A and C.

For what concerns the behavior of the models, it's easy to see that in all the countries, the ARIMA has underestimated the price with respect to the GRU and the LSTM. In reality, this is a direct consequence of the fact that the k_t is lower in the case of the ARIMA.

This aspect is very important in annuities calculation, in fact, the consequences in estimating a lower price with respect to the real is that the company will face big loss

when they will have to pay more than expected. While, in the case of overestimating the price, the problem will be that the company will ask to the policyholder more than it should and this will mean that the company could be not competitive on the market.

The most interesting is to compare the benchmark model, which is constructed using the real probabilities, to the other models in the annuity type B.

In general, it could be observed that the lowest differences between the model and the benchmark is reached in the model that has the lowest Root Mean Squared Error and Mean Absolute Error. In all the countries, the two indicators are in accordance, except for the Italian case. In facts according to the RMSE the best model is the LSTM, while looking at the MAE the best model is the GRU. In this case, the best model is the GRU, since the differences in the case of the MAE have a higher magnitude with respect to the RMSE.

In this context, it can be seen that the GRU architecture outperforms the other models on three of the four countries and for this reason, it could be considered the best model above the others. Sometimes the GRU and the LSTM models are quite similar considering type C annuities, anyway it's due to the facts that in the age considered for these annuities, the models behave similarly in terms of k_t.

The differences between the pricing of the models are higher in the annuities of type A since there is a higher time interval to be considered and so, higher forecasted probabilities.

For what concerns the differences inter models in term of pricing, the highest change is observed in the type A annuity on the Australian country. In facts, the LSTM has a 30% difference with respect to the ARIMA, while the GRU a 18% difference. In facts also the differences in terms of RMSE are the highest in this country looking at the different models.

Comparing the prices of the LSTM and the GRU, it's possible to observe that the prediction are similar but looking at the contracts of type A and B since we don't have the future probabilities, it's not possible to say which is more precise and we should limit our critical analysis to the annuities of type C.

In general, all the models underestimate the real price of the annuity, that is what it has been observed in the probabilities of die, where are in most of the cases overestimated. This is due to the fact that the k_t is underestimated from all the models in general and also that there is an error that is implicit in the use of the Lee-Carter model.

Concluding remarks

This master thesis has been focused on the thematic of the mortality. This argument is important and fundamental in actuarial science fields, in particular, in life insurance context and pension funds. In addition to this, mortality is studied in demography that is the statistical study of human populations with reference to size and density, distribution, and vital statistics.

In the recent years, a lot of application based on Neural Networks models have been proposed in the actuarial sciences field and more specifically the mortality modelling. In my opinion these researches are very interesting. For this is the reason, I have decided to approach this argument in my master thesis.

The analysis that I've conduced discuss the integration of the Recurrent Neural Networks to the Classical Lee-Carter model and the comparison of the results with the Classical Lee-Carter scheme. The Lee-Carter model in this analysis has been developed with the Singular Value Decomposition.

The first point that is coming out from this analysis is the improvement of the mortality in the time. This is a fact, in most of the countries in the world, the more we go further in the history, until here, the mortality has improved, as it has been shown in chapter 2.

For this reason, it has seemed to be reasonable that the forecast obtained from the Lee-Carter model application has shown a general continuous improvement of the length of the life in the time.

But how much is improving in the time? This is a matter of actuarial and mathematical calculation. In facts, different models can be applied to solve this issue, but they give different outcomes. The model which has been taken into account is the well-known Lee-Carter model, which is one of the traditional models that it's used to forecast the mortality over the time. This model projects the mortality in the future thanks to the projection of the time dependent factor k_t. The experiments that have been carried in this thesis can be subdivided into two macro categories: traditional ARIMA model and a deep learning integrated Lee-Carter model using Long-Short Term Memory and Gated Recurrent Unit architectures.

The results are clear, the deep learning integrated Lee-Carter model overcome the canonical ARIMA Lee-Carter model in terms of minimum error on the considered test set. The result is validated also by comparing the outcomes in terms of death probability with respect to the real probability of death.

In facts, the ARIMA Lee-Carter model tends to create a trivial shape of forecasted trend and forecasted mortality over time. In contrast, a deep learning integrated Lee-Carter model is capable to find a more interesting and realistic non-linear behavior. The approach of the neural network is able to catch, memorize and then replicate in a more accurate way the trend that is inside the k_t factor (*Negri et al 2019*).

The possibility to find a more accurate estimation of the survival probabilities in the case of the Neural Networks model is very important and appreciated in the contest of pension plan, social security scheme. In facts, in this way, policies which are considering long term cashflows are able to use the Lee-Carter model in a more flexible way and obtain a more accurate price and forecasts. As it has been demonstrated in the chapter 5, the differences in the price of annuities are very relevant comparing ARIMA and Neural Network models In fact, the RNN reduce the underestimation of the survival probabilities and this can help companies in reducing the risk of loss in the future due to the underestimation of this factor.

The comparison between the Gated Recurrent Unit and the Long-Short Term Memory doesn't show relevant differences, but it's interesting to notice that GRU architecture has demonstrated more accurate and stable results.

An important thing to notice is the fact that, differently from the ARIMA model, the Neural Networks are providing a point estimate. The analysis of the variability, and the production of an interval of confidence of the prediction is a challenge in this field. There are some scholars that have tried to face this problem, but nobody has approached it in time series analysis, providing a confidence interval and a volatility measure of the series.

It's relevant to point out that the analysis carried out in this thesis only considers the so called "Classical Lee-Carter Model": the method that has been developed by *Ronald D. Lee and Lawrence R. Carter (1992).* This model utilizes the Singular Value Decomposition to decompose the matrix of age specific mortality rates.

This methodology suffers a lot of limitations as showed by *Ronald D. Lee (2000)*. In the practice a lot of different extensions of the Classical Lee-Carter and other methodologies can be used to project the mortality. Some of these extension regards the parameters decomposition, such as the Weighted Least Squares suggested by *Wilmoth (1993)*, the Maximum Likelihood Estimation proposed by *Brouhns et al. (2002)*, and they can provide a more efficient estimation. Other models have improved the Traditional Lee-Carter model introducing additional effects, as for example the cohort dependent component, that is not considered in the Classical Lee-Carter, this method has been proposed by *Renshaw and Haberman (2006)*. All these models have different performances and the criteria through which the right method should be chosen can depend on different factors. Nevertheless, the scope of this master thesis has been limited to the Classical

model, and the results have to be considered valid only in this specific context considered.

In conclusion, this thesis has covered only partially the study of Neural Network in mortality theory and there can be a lot of further extensions that can be created and proposed. One interesting point could be to use other Neural Network models, such as Convolutional Neural Network in order to provide the forecast of the k_t time series.

In additions could be interesting to try to model the k_t time series with some manipulation, maybe trying to make the series stationary and then use a Neural Network to provide the forecast as it has been done in some recent Neural Networks time series projection and analysis. Or as opposite try to forecast directly the death rates directly, using a time series per age and years.

Finally, on of the most interesting expansion of the model could be the construction of a volatility measure and the confidence interval for the Neural Network forecast. In this way, companies could be able to have a quantitative risk measure associated to the point estimate, and this could be really helpful for the insurers.

List of figures

List of tables

Bibliography

- *Alpaydin E. - Introduction to Machine Learning - 2nd ed. The MIT press – 2010*
- *Bianchi F. M., Maiorino E., Kampffmeyer M. C., Rizzi A., Jenssen R. - Recurrent Neural Networks for Short-Term Load Forecasting – Spinger - 2017*
- *Blanchard R. - Getting started with deep learning using the SAS Language – 2020*
- *Bodén M. - A guide to recurrent neural networks and Backpropagation - 2002*
- *Brouhns N., Denuit M., Vermunt J. K. - A Poisson log-bilinear regression approach to the construction of projected lifetables - Insurance: Mathematics and Economics Vol.31 - 2002*
- *Cairns A. J. G., Blake D., Dowd K. – A two factor model for stochastic mortality with parameter uncertainty: theory and calibration - The Journal of Risk and Insurance, Vol. 73 - 2006*
- *Chen G. - A Gentle Tutorial of Recurrent Neural Network with Error Backpropagation - 2018*
- *Chung J., Gulcehre C., Cho K., Bengio Y. - Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modelling - 2014*
- *Danesi I. L.- Forecasting Mortality in Related Populations Using Lee-Carter Type Models – 2014*
- *Debonneuil E. - A simple model of mortality trends aiming at universality: Lee Carter + Cohort - AXA Cessions, Paris, France – 2010*
- *Denuit M., Hainaut D., Trufin J. - Effective Statistical Learning Methods for Actuaries III Neural Networks and Extensions – Spinger - 2010*
- *Deprez P., Shevchenko P. V., Wüthrich M. V. - Machine learning techniques for mortality modeling - European Actuarial Journal Vol. 7 - 2017*
- *Efron B., Hastie T. - Computer Age Statistical Inference Algorithms, Evidence, and Data Science - Stanford University – 2017*
- *Friedman J. H., Tibshirani R., Hastie T. - The Elements of Statistical Learning: Data Mining, Inference, and Prediction: Data Mining, Inference, and Prediction - 2nd ed. Spinger – 2013*
- *Gers F. A, Schraudolph N. N., Shmidhuber J. - Learning Precise Timing with LSTM Recurrent Networks - Journal of Machine Learning Research Vol.3 - 2002*
- *Gers F. A. Schmidhuber J. - Recurrent nets that time and count - International Joint Conference on Neural Networks - 2000*
- *Hainaut, Donatien - A neural-network analyzer for mortality forecast - The Journal of the IAA vol. 48- 2018*
- *Hyndman R. J., Koehler A. B. - Another look at measures of forecast accuracy - International Journal of Forecasting vol. 22 – 2006*
- *Hyndman R.J., Athanasopoulos G. - Forecasting: Principles and Practice –Monash University, Australia -2018*

- *James G., Witten D., Hastie T., Tibshirani, R. - An Introduction to Statistical Learning with Applications in R – 2017*
- *Kingma D. P., Lei Ba J. – ADAM: A method for stochastic optimization - ICLR - 2015*
- *Koissi M. C., Shapiro A. S. - The Lee-Carter model under the condition of variable age-specific parameters - 43rd Actuarial Research Conference, Regina, Canada - 2008*
- *Laplace P. S. - A Philosophical Essay on Probabilities – 1825*
- *Lavda F. - A Study of Recurrent neural networks (RNNs) in univariate and multivariate time series - 2017*
- *Li L., Jamieson K., DeSalvo G., Rostamizadeh A., Talwalkar A. - Hyperband: a novel bandit-based approach to hyperparameter optimization - 2017*
- *Lee R. D.- The Lee-Carter method for forecasting mortality, with various extensions and applications - North American Actuarial Journal vol.4 - 2000*
- *Lee R. D., Carter L. - Modeling and Forecasting the Time Series of U.S. Mortality - Journal of the American Statistical Association vol.87 - 1992*
- *Levantesi S., Menzietti M. - Giornata degli attuari delle pensioni, Gruppo di lavoro dei percettori pensioni/rendite –sede INPS, Roma - 2011*
- *Levantesi S., Pizzorusso V. - Application of machine learning to mortality modeling and Forecasting - Risks vol.7- 2018*
- *Macdonald A. S., Richards S. J., Currie I. D. - Modelling Mortality with Actuarial Applications – Cambridge University press - 2018*
- *McCulloch W. S., Pitts W. - A logical calculus of the ideas immanent in nervous activity - Bulletin of Mathematical Biophysics vol. 5 - 1943.*
- *Negri, S. Levantesi, M. Marino, S. Scognamiglio, F. Perla - A Deep Learning Integrated Lee-Carter Model -Risks vol.7 – 2019*
- *Olah C. – "Understanding LSTM Networks" – colah's blog -2015*
- *Olivieri A., Pitacco E. - Introduction to Insurance Mathematics Technical and Financial Features of Risk Transfers - 2nd ed. Spinger - 2010*
- *Olivieri A., Pitacco E., Haberman S., Donuit M. - Modelling Longevity dynamics for pensions and annuities business - Oxford University press – 2009*
- *Paci L. – Empirical Research course notes – a.y. 2019 - 20*
- *Pascanu R., Gulcehre C., Cho K., Bengio Y. - How to Construct Deep Recurrent Neural Networks - 2013*
- *Pascanu R., Mikolov T., Bengio Y. - On the difficulties of training a Recurrent Neural Networks – 2013*
- *Pascarius M. D. – Modelling and forecasting mortality - 2018*
- *Perla F., Richman R., Scognamiglio S., Wüthrich M. V. - Time-Series Forecasting of Mortality Rates using Deep Learning - 2020*
- *Petnehàzi G., Gall J. - Mortality rate forecasting: can recurrent neural networks beat the Lee-Carter model? - 2019*

- *R. Richman, M. V. Wüthrich - Lee and Carter go Machine Learning: Recurrent Neural Networks - Swiss Association of Actuaries - 2019*
- *Richman R., Wüthrich M. V. - A Neural Network Extension of the Lee-Carter Model to Multiple Populations - Annals of Actuarial Science vol.1 - 2018*
- *Ronald Richman - AI in Actuarial Science - ASSA Convention, Cape Town - 2018*
- *Rosina A., De Rose A.- Demografia – EGEA, 2nd ed. - 2017*
- *Ruder S. - An overview of gradient descent optimization algorithms - Insight Centre for Data Analytics, NUI Galway – 2017*
- *SAS Institute - Deep Learning using SAS® Software - Course - 2019*
- *SAS® Deep Learning Action Set - support.sas.com/en/documentation.html*
- *Sepp H., Schmidhuber J. - Long Short-Term Memory - Neural Computation vol.9 - 1997*
- *Spedicato G. A., Clemente G. P. - Mortality projection with demography and lifecontingencies packages - cran.r-project.org*
- *Wilmoth J. R. - Computational methods for fitting and extrapolating Lee-Carter model of mortality change – 1993*
- *Zadranská L. – Time Series Forecasting using Deep Neural Networks – 2019*
- *https://docs.paperspace.com/machine-learning/wiki/activation-function*
- *www.actuarialdatascience.org*
- *www.cran.r-project.org*
- *www.istat.it*
- *www.machinelearningmastery.com*
- *www.mortality.org*
- *www.rdocumentation.org/packages/demography*
- *www.rdocumentation.org/packages/forecast*
- *www.towardsdatascience.com*

Acknowledgement

This thesis has been written as final elaborate of my Master of Science in Statistical and Actuarial Sciences. This path has been very significant and a source of enrichment for me: I've had the opportunity to make a lot of experiences thanks to the Professors, the Faculty and my own curiosity.

During this course I learned the importance of the data and that the risks and decisions that we will experience in future can be measured in a quantitative way, in present terms. Mortality is one of the risks that everyone faces every day, this phenomenon has a huge impact at different levels: for each individual, for the whole society. This topic has fascinated me and for this reason I decided to focus on this theme in my master thesis.

I'm very much thankful to the professor Gian Paolo Clemente for having supported my curiosity for this theme and for his constant guidance during the research.

During the summer break of the academic year 2018-2019 I've been to the Aarhus University (Denmark) to attend a course in Data Science in Insurance. This multicultural context has given me the opportunity to interact with people from all the Europe that had my same academic interest but from different prospects. I thank the Professor Katrien Antonio for the passion and the energy that she has transmitted during this experience.

The Catholic University of Sacred Heart and the Faculty of Banking, Finance and Insurance studies has given me the opportunity to enrich my knowledge and experience thanks to the organization of digital skills courses and the "Talent Project", that I've attended. I'm very grateful for having been selected for this initiative that have let me the opportunity to reach a good knowledge of SAS software and obtain the SAS programmer certification. In this context, I want to thank Cinzia Gianfiori, Francesca Sciloretti Diana for the tutoring during the "Talent Project" and Christopher Minafra for the mentoring during my work experience at SAS Institute that has followed the project.

This study path has been very intense and rich of emotions, I want to thank my family, my parents Giuseppe and Luigia, and my sister Veronica, for having let me the opportunity to attend this course and for the continuous support and understanding.

I want to thank my grandfather Vincenzo for the constant encouragement, I wish you were here to enjoy this special moment with me.

I want to express my gratitude to Ilaria, the girl that has accompanied me during this path, her support and closeness during the moments of joy and difficulty has been extremely important for me.