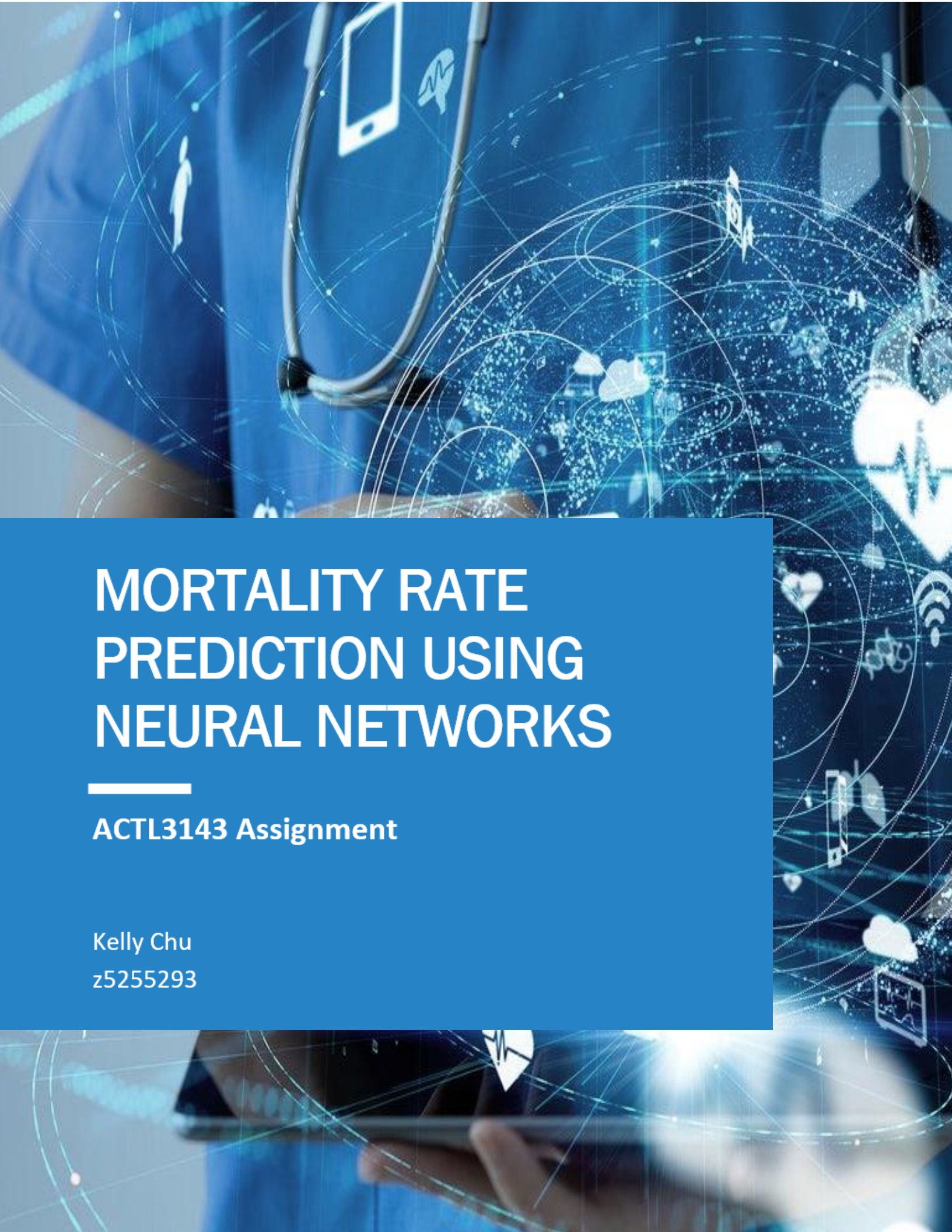


MORTALITY RATE PREDICTION USING NEURAL NETWORKS

ACTL3143 Assignment

Kelly Chu
z5255293

Contents

Introduction

The ability to quantify and interpret mortality rates is central to demography studies of the human population. Within an actuarial context, mortality rates are basic inputs in models such as those for the pricing and valuation of life insurance products. Since mortality rates evolve over time, accurate models to forecast future mortality rates is central to understanding drivers of change and establishing sound assumptions for actuarial calculations.

The project will use past Australian population data to predict future mortality rates. This is a regression type problem, regressing central log mortality rates $\log(m_x)$ against calendar year t, age x and gender to forecast mortality rates. A baseline Lee-Carter model is fitted before looking to how predictions can be improved with two deep learning models – deep and wide network and long short-term memory (LSTM) network.

The Data

The mortality dataset is based on data from the Human Mortality Database (HMD). For the purposes of this project, the Australian population data, labelled "**AUS**" in HMD, has been selected but the study can be easily generalized to any country in the database.

As per Fig 1, pre-processing steps were performed to ensure the input is in the appropriate format for the application of the models to be fitted. Consequently, the study considers both women and men from age 0-99 between the years 1921-2019. A summary of the raw and cleaned dataset has been included in A1 and A2.

Fig 1:Summary of Key Data Pre-Processing

Variable	Step	Explanation
All	Melt dataframe	Variables to be considered to include year, age, gender set to be dataframe columns
Age	Encode age bracket "110+"	Allows variable type can be changed to Integer
	Cap maximum age to 99	Deals with missing and sparse data of old ages
m_x	Replace missing values "." and "0" with NA – removed when capping age	Logarithm cannot take inputs of 0 or text. Mortality rates must be strictly positive.
	Log m_x	Input and output of LC model. For consistency, use logarithm to compare between all three models

Fig 2:Data Stratification

To fit the model, the data is stratified by calendar year, as defined in Fig 2. Using the training set, each model was trained and hyperparameters tuned until optimal, and then assessed on the validation set. The model with the lowest validation error was then fitted on the test set.

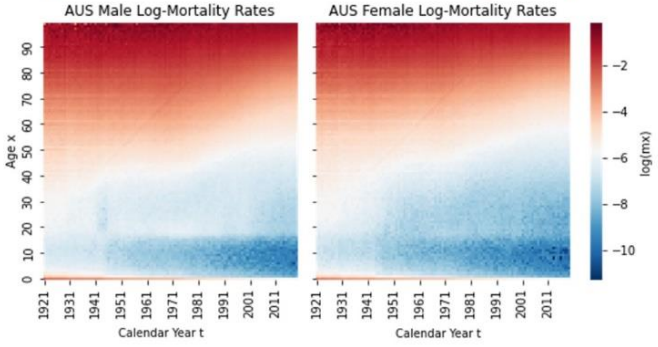
Exploratory Data Analysis

EDA of the mortality dataset reveals differences in mortality rates between genders, ages and over time. Figure 3 uses a heatmap to illustrate the log-mortality rates for both the Australian male and female populations. Here, the gradual color change from red to blue represents decreases in log-mortality rates.

The following insights can be observed:

- Mortality rates for females are lower than for males.
- The upward-sloping diagonal pattern reflects improvements in mortality rates over time.
- Age effects can be shown through the vertical patterns in the heatmap. For a given year, mortality rates tend to increase as age increases.
- Period effects can be shown through the horizontal patters in the heatmap. For a given age, mortality rates tend to decrease as calendar year increases.

Fig 3: Heatmaps of Log-Mortality Rates of AUS Male (LHS) and AUS Female (RHS) population from calendar years 1921 to 2019 and for ages 0 to 99 years



These observations can be further highlighted by the further EDA performed in **A3**.

Baseline Model – Lee Carter Model

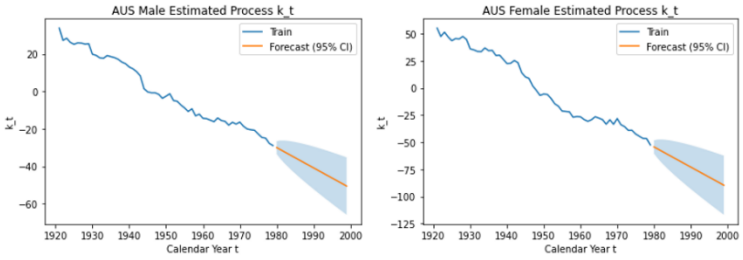
The Lee-Carter model is one of the most used models for mortality forecasting since its introduction in 1992 (Lee, Carter 1992). The model is based upon the below equation where for a given age x in the calendar year t, the logarithm of the central mortality rate can be estimated as:

$\log \hat{m}_{t,x} = a_x + b_x k_t$	where: $m_{t,x}$ = central death rate at age x in year t a_x = average age-specific pattern of mortality b_x = age-specific patterns of mortality change as k_t varies k_t = time index describing mortality trend over time

As per the original Lee and Carter approach, the parameters are fit on the training dataset using Singular Value Decomposition. The process of deriving the optimal parameter values is outlined in A4. As a single population modelling approach, the model is fitted separately for females and males.

Since a_x and b_x are both age-dependent, it is assumed that they are constant over time. Hence, to project mortality rates beyond the training set, k_t is the only parameter required to be extrapolated. k_t is thus modelled as a random walk with drift, represented as an independent ARIMA (0, 1, 0) process. The forecast is shown in Fig 4.

Fig 4: Estimated ARIMA (0, 1, 0) Process k_t



Using the fitted parameters, we then obtain the predicted $\log \hat{m}_{t,x}$ as per the above equation and compare the predictions with the observed mortality rates to calculate the mean-squared error (A5). The results for the Lee Carter model are provided in Figure 5.

Fig 5: LC Model Training and Validation MSE Losses

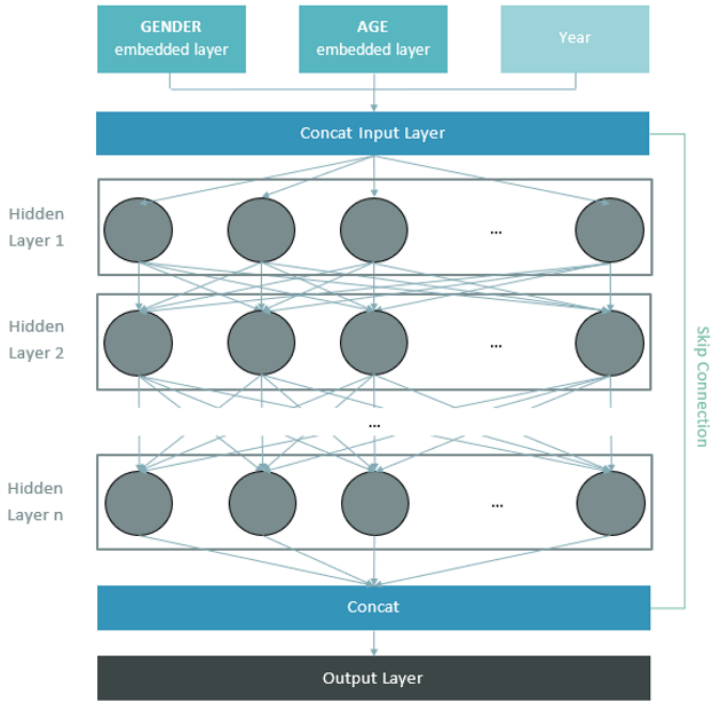
	Male	Female	Both
Training MSE	4.35	5.97	5.16
Validation MSE	3.41	11.84	7.62

All values have been multiplied by 10^4

Neural Network #1 – Deep and Wide

The Model

Fig 6: Graphical Depiction of the Deep and Wide Model



Rather than relying on a specific functional form like the LC model, neural networks (NNs) provide highly flexible regression models that can learn from the features of data by using calendar year, gender and age as predictors. The architecture of the deep and wide model to be fitted is shown in Figure 6. This structure combines the benefits of memorization provided by the wide layers and generalization by the deep structure.

As only numerical inputs can be fed into NNs, gender and age (which is interpreted as a categorical variable) must be treated. An embedding layer can be used to map each category to a low dimensional vector and these embeddings can be learned by the neural network for network calibration (Schnürch and Korn, 2022). The embedding layer and calendar year (numerical scaled input) is then concatenated to be used as the NN input.

Layer by layer of interconnected nodes, NNs then use a series of algorithms to recognize relationships between the data by optimizing the weights and biases in each layer using a selected activation function. This process is outlined in A6.

For simplicity, the diagram does not include the batch normalization and dropout layers. An aspect of NN training is that the model is updated by back propagation which assumes that the weights in prior layers to the current are fixed. However, as explored by Ioffe and Szegedy (2015), the distribution of each layer's inputs changes during training as the parameters of the previous layers change. As such, batch normalization can be used to standardize the inputs to each layer which can help coordinate the update of multiple

layers in the model to improve the stability and training time of the NN. To prevent over-fitting, dropout layers was also be used to randomly set input units to 0 with a set probability. This regularizing technique is computationally cheap but can improve generalization error in the NN.

Hyperparameter Tuning

Model architecture components can be optimized to improve the predictive power of the model. The search space and description for each hyperparameter to be tuned is summarized in Fig 7. Hyperparameter tuning was performed on the training data using Bayesian Optimizer with 20 trials (i.e., 20 combinations of hyperparameters were considered). The top five models are listed below in Fig 8 – the model with the lowest validation error was selected for model comparison. The architecture of the selected deep and wide model is explored in A7.

Fig 7: Hyperparameter Search Space for Deep and Wide Network

Hyperparameter	Description	Search Space
Learning Rate	Controls how quickly network updates its parameters	0.001, 0.01, 0.1
Number of hidden layers	Number of layers between input and output layer	1, 2, 3, 4
Number of neurons per hidden layer	Small number can produce underfitting whilst too many neurons can produce overfitting since network learns too much from training data and doesn't generalise	10, 20, 30, … ,100
Dropout Probability	Form of regulatorisation to avoid overfitting where random neurons are cancelled given a specified probability	0 (no dropout), 0.1, 0.5
Size of age and gender embedding	Number of dimensions of vector to map category to	2, 3, 5
Activation function	Allows deep learning models to learn non-linear prediction boundaries	Relu, Tanh
Number of epochs	Number of times the whole training data is shown to the network while training	Max 100 with early stopping (patience 10)

Fig 8: Top 5 Deep and Wide Models and Their Hyperparameters

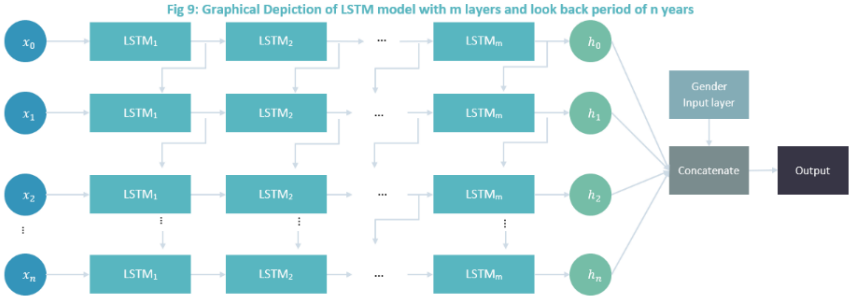
Learning rate	Number of layers	Neuron Architecture	Activation Function	Dropout Probability	Age & Gender embedding	Loss Score
0.001	4	10, 100, 100, 10	Tanh	0	2, 3	0.031526
0.001	4	90, 100, 70, 100	Tanh	0	5, 5	0.031619
0.001	4	10, 100, 100, 40	Tanh	0	5, 2	0.031621
0.001	4	100, 100, 10, 100	Tanh	0	5, 2	0.032756
0.001	4	100, 100, 100, 100	Relu	0.02	5, 3	0.032869

Neural Network #2 – LSTM

The Model

A weakness of the deep and wide model is that the time series structure can get lost as the model is not designed to respect time and causal relationships (Richman, Wuthrich, 2019). This is addressed through recurrent neural networks which link the hidden layers of the network cyclically so that values computed depend on previous parts of the sequence (Fig 9).

Fig 9: Graphical Depiction of LSTM model with m layers and look back period of n years



However, for a longer sequence, RNNS may suffer from short-term memory where the network can forget what has been seen in earlier portions of the sequence. This is addressed by LSTM networks which can learn to keep only relevant information to make predictions and forget non-relevant data through a system of sub-networks called gates (explored in A8). We also layer an additional binary indicator for the genders of the observations so that the model can predict jointly for both genders.

Data Processing

Input to the LSTM model requires conversion of the time series of mortality rates into subsequences of a selected look-back period of n years, using a rolling window approach. As such, the input of the network is a matrix of size [number of ages considered × (number of years considered - n), n, 1]. This approach implicitly treats age as another observation of the time series rather than directly feeding age as a numerical / categorical variable. The log-mortality rates were also standardised using Min-Max Scaling.

Hyperparameter Tuning

The search space for each hyperparameter to be tuned is summarized in Fig 10. Bayesian Optimizer was then used with the top 5 models are shown in Fig 11. The selected model with the lowest validation error is explored in A9.

Fig 10: Hyperparameter Search Space for LSTM Model

Hyperparameter	Search Space
Learning Rate	0.001, 0.01, 0.1
Number of hidden layers	1, 2, 3
Number of neurons per hidden layer	5, 10, 15, 20
Activation function	Relu, Tanh

Fig 11: Top 5 LSTM Models and Their Hyperparameters

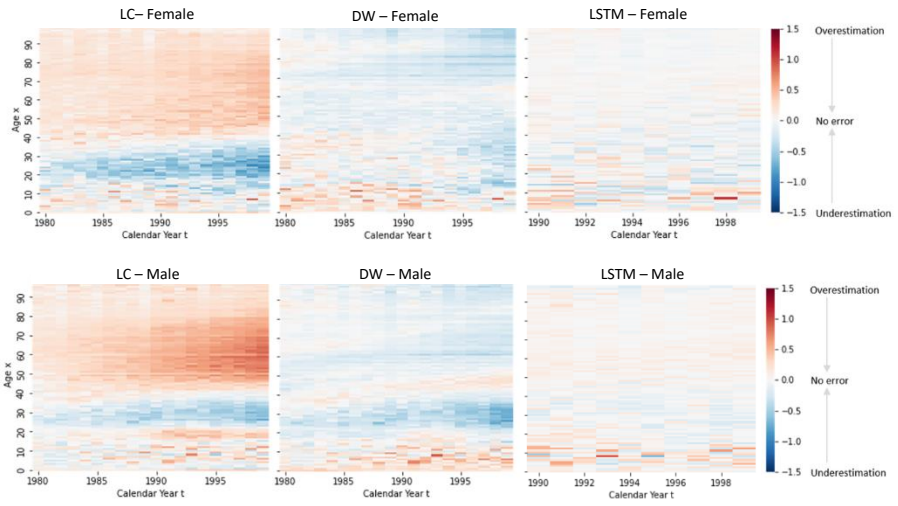
Lookback Period	Learning rate	Number of layers	Neuron Architecture	Activation Function	Loss Score
10	0.001	2	20, 5	Relu	0.017935
10	0.001	2	5, 20	Tanh	0.017994
10	0.001	2	20, 10	Tanh	0.018027
10	0.001	2	20, 10	Relu	0.018491
10	0.001	2	20, 5	Relu	0.018589

Model Comparison

After fitting the model on the training set, the models are then used to predict on the validation set and assessed.

The residuals of the predictions compared to the observed were calculated and depicted in a heatmap (Fig 12). The color map reveals the size of the residuals with a large positive residual (overestimation) being represented by red whilst blue represents a negative residual. As such, it can be observed that the LSTM model performs the best overall as it produces smaller residuals overall.

Fig 12: Heatmap of Residuals on Validation Dataset Predictions



Compared to the DW network and LC model, the heatmap reveals that the LSTM model has learned both age and period effects since there are no obvious patterns in the residuals. Contrastingly, it can be observed that the LC model overestimates mortality for younger and old ages and underestimates for middle ages. This is less pronounced in the DW residual heatmap. This is further analysed in A10 which supports our conclusion that the LSTM model has the higher predictive power compared to the other two models across all ages and calendar years. This is likely attributed to the LSTM architecture allowing for previous outputs to be used as input, thus maintaining the relationship between time steps which is useful for time series applications.

Across genders, it can be observed that residual patterns are more pronounced when using the LC model to predict for males whilst the DW model tends to perform better for females as the residual pattern for middle ages is less pronounced. The LSTM model appears to perform well across both genders.

As shown in Fig 13, it can be observed that the LSTM model has the best performance as it has the significantly lowest MSE score across all models and on both the training and validation sets. Since the LSTM model has the lowest validation error, it was then fitted on to the test set.

In units of 10^{-4}, the test MSE was 0.089 for females and 0.314 for males with a combined MSE of 0.2018. The miniscule MSE loss suggest the overall good predictive power of the LSTM for the mortality forecasting problem.

This is reflected in Figure 14 which shows that predictions lie close to the 45-degree line in a plot of predictions against actual values. Time trends are captured well with residuals averaging 0 and demonstrating no visible patterns, it can be seen the LSTM model is not as strong for younger demographics. This may be due to the higher volatility of younger age mortality rates which is relatively more difficult to be captured with a model that is applied to all ages.

Fig 13: MSE in 10^{-4} Comparison

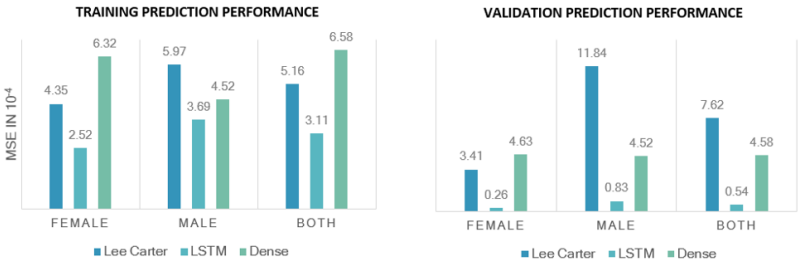
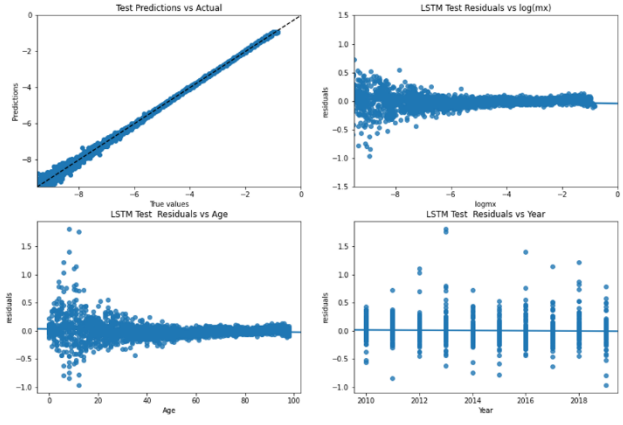


Figure 14: LSTM Predictive Performance on Test Set



Ethical Considerations

Data Source

A key ethical concern of any data analytics project is the data source and the degree of individual privacy security. The dataset used is in de-identified aggregated form which prevents any sensitive information from being leaked. For the Australian dataset, the data source is the government agency, ABS, which has a commitment to "open transparent use of personal information" through stringent privacy policies and secure IT systems. As such, the data is compiled from a secure and reliable source which does not infringe any personal information privacy rights.

Importance in risk-based pricing and valuations

Central mortality rates are often used in the assumption setting process of various actuarial models as it can give an indication of the likelihood of death and the expected lifetime of an individual. Some applications include the following:

1. Life Insurance Pricing	If an individual has a higher probability of death, the likelihood of a claim increases. To account for increase risks, they may be charged a higher premium.
2. Annuity Pricing	If an individual has a lower probability of death, expected lifetime is longer. To account for longer expected period on the annuity, the cost of the annuity is higher.
3. Reserving	If portfolio has higher probability of the covered event happening (e.g. death claim), insurance company have to build up reserves to be prepared for increased risks.

Accurate assumptions of mortality are important to ensure the pricing and reserving of product to reflect the risks involved. Improved forecasting can help insurers more accurately account for and manage risks to improve profitability and solvency.

Price Discrimination

Our study has highlighted the significant relationship between age, gender and calendar year (effectively, birth year is also considered which is the difference between calendar year and age). Understanding of these driving factors of mortality can reduce underwriting risks. This is because, at its core, the insurance industry is built upon risk classification through its ability to differentiate among risks and group insureds into homogenous classes for the purpose of actuarial equity.

However, such implementation of death predictors can challenge risk-pooling principles which many insurance products are built on and can lead to price discrimination against individuals. For example, if mortality rates are expected to be higher for male, is it equitable to charge a male higher insurance price or would it classify as gender discrimination? Likewise, similar discussions occur when considering age as charging more expensive premiums for older members who generally face higher risk of death may make insurance products unaffordable, particularly towards the end stages of the financial lifecycle.

Consequences for Future Studies

Although this study was on a more aggregated level, the high predictive power of the model validates that artificial intelligence can be used to identify further, even more individualized features of a person to predict prognosis. This is reflected on further studies that look at mortality on a more granular scale with factors such as disabilities and race which have shown linkages to mortality outcomes.

Although this can bring benefits as outlined above, there must be ethical considerations of the consequences of discrimination and its impact on the availability and affordability of insurance coverage if identified variables were incorporated as rating factors. Solutions designed from the findings by artificial intelligence models must balance bias and accurate differentiation amongst risks.

Appendix

A1. Raw Dataset Summary

	Year	Age	Female	Male	Total
0	1921	0	0.059987	0.076533	0.068444
1	1921	1	0.012064	0.014339	0.013225
2	1921	2	0.005779	0.006047	0.005916
3	1921	3	0.002889	0.004197	0.003554
4	1921	4	0.003254	0.003254	0.003254
...	...	...	...	...	...
10984	2019	106	0.566844	0.591484	0.570138
10985	2019	107	0.591219	0.748423	0.608240
10986	2019	108	0.630787	1.628866	0.683836
10987	2019	109	0.751503	.	0.751503
10988	2019	110+	1.508079	.	1.508079

10989 rows × 5 columns

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10989 entries, 0 to 10988
Data columns (total 5 columns):
 #   Column  Non-Null Count  Dtype 
---  ------  --------------  ----- 
 0   Year    10989 non-null  int64 
 1   Age     10989 non-null  object
 2   Female  10989 non-null  object
 3   Male    10989 non-null  object
 4   Total   10989 non-null  object
dtypes: int64(1), object(4)
memory usage: 429.4+ KB
```

A2. Cleaned Dataset Summary

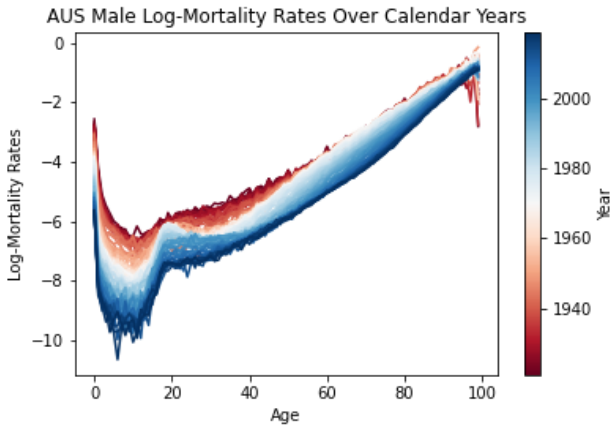
	Year	Age	Gender	mx	logmx
t					
1921-01-01	1921	0	Female	0.059987	-2.813627
1921-01-01	1921	1	Female	0.012064	-4.417529
1921-01-01	1921	2	Female	0.005779	-5.153525
1921-01-01	1921	3	Female	0.002889	-5.846845
1921-01-01	1921	4	Female	0.003254	-5.727870
...	...	...	...	...	...
2019-01-01	2019	95	Male	0.273874	-1.295087
2019-01-01	2019	96	Male	0.293684	-1.225251
2019-01-01	2019	97	Male	0.346880	-1.058776
2019-01-01	2019	98	Male	0.373261	-0.985477
2019-01-01	2019	99	Male	0.406094	-0.901171

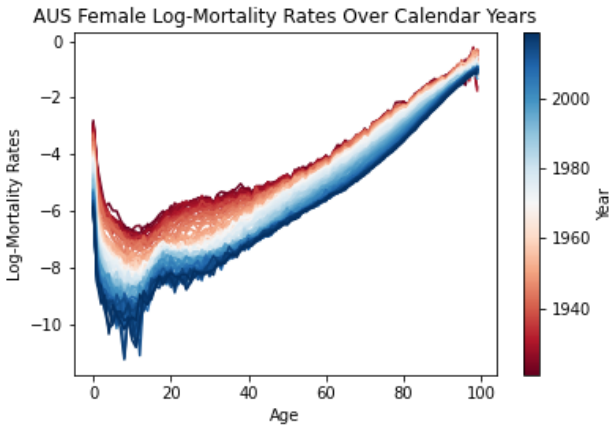
19800 rows × 5 columns

```
<class 'pandas.core.frame.DataFrame'>
DatetimeIndex: 19800 entries, 1921-01-01 to 2019-01-01
Data columns (total 5 columns):
 #   Column  Non-Null Count  Dtype   
---  ------  --------------  -----   
 0   Year    19800 non-null  int64   
 1   Age     19800 non-null  int32   
 2   Gender  19800 non-null  category
 3   mx      19800 non-null  float64 
 4   logmx   19800 non-null  float64 
dtypes: category(1), float64(2), int32(1), int64(1)
memory usage: 715.6 KB
```

A3. Further EDA

Patterns in mortality at different ages reflect the human lifecycle.



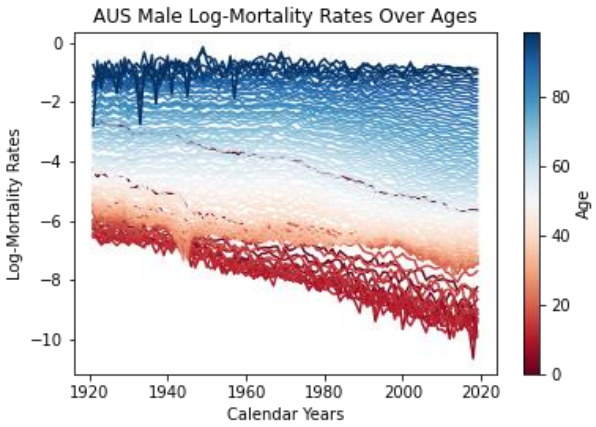


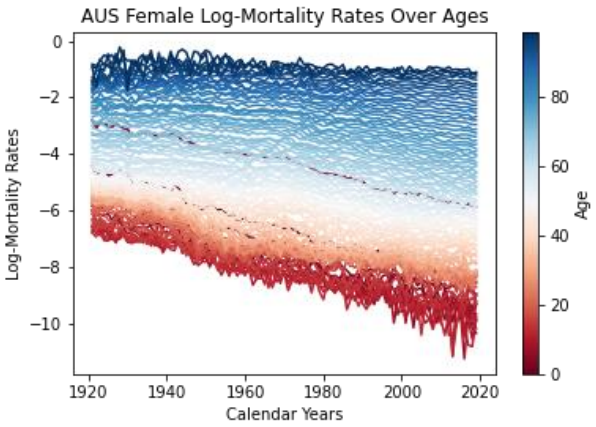
Higher mortality in newborn / infants which decreases over childhood

Increase over teenage years until trough in mid 20s

Increase into high mortality of the elderly

In the below plot, we can observe that improvements in mortality over the past decades is more evident in younger ages - particularly of infants and children whereas those in older ages have experienced relatively less mortality improvements.





A4. Deriving the Parameters of the LC Model

The Lee-Carter model is the most used model for mortality forecast since its introduction in 1992. It extrapolates trends and age patterns in mortality data to forecast future mortality rates.

Suppose we are considering ages x = 1,2,...,N and calendar years t = 1,2,...,T.

The model calculates the logarithm of the central death rate $\log m_{t,x}$ at age x in the calendar year x as:

$$\log \hat{m}_{t,x} = a_x + b_x k_t$$

where:

- $m_{t,x}$ = central death rate at age x in year t
- a_x = average age-specific pattern of mortality
- b_x = age-specific patterns of mortality change as k_t varies
- k_t = time index describing mortality trend over time

We will calculate the parameters from first principles. The model aims to find the least squares solution to equation 1. with the following procedure to estimate the parameters:

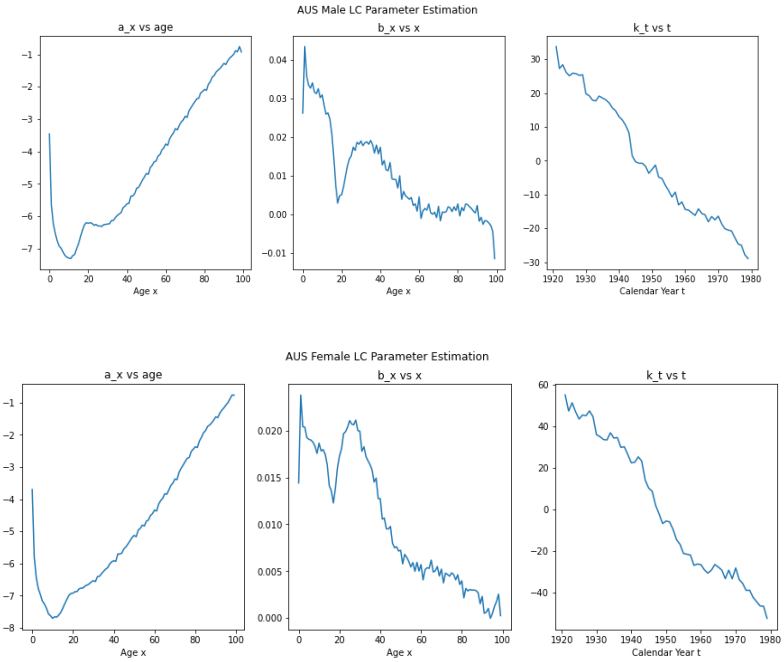
1. Set $\tilde{a} = \frac{\sum_{t=1}^{T} \log m_{x,t}}{T}$
2. Center the raw log-mortality rates to get $\tilde{M}_{x,t} = \log m_{x,t} - \tilde{a}$
3. As Lee and Carter suggests, use Singular Value Decomposition of $\tilde{M}_{x,t}$ where U, S and V represents the age component, singular values and time component respectively
$$\text{svd}(\tilde{M}_{x,t}) = USV^T$$
4. Solving this provides our estimates: $\tilde{b}_x = U_{x1} S_1$ and $\tilde{k}_t = V_{t1}$
5. To ensure model identifiability (i.e. to obtain a unique solution), we then re-scale the estimates to satisfy the constraints $\sum_x \hat{b}_x = 1$ and $\sum_t \hat{k}_t = 0$. As such, the estimates are as follows:

$$\hat{a} = \tilde{a} + c_1 + \tilde{b}_x$$
$$\hat{b}_x = \frac{\tilde{b}_x}{c_2}$$
$$\hat{k}_t = \tilde{k}_t$$

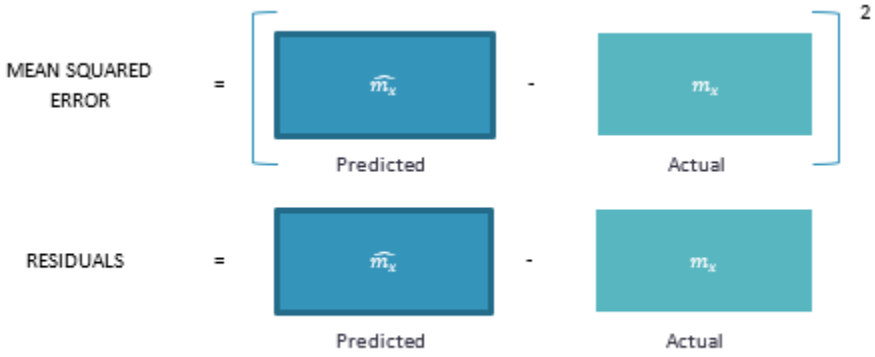
Where:

- c_1: mean of $\tilde{k}_t$
- c_2: sum of $\tilde{b}_x$

The derived parameters (plotted below) reflects much of the trends explored in the initial EDA phase.

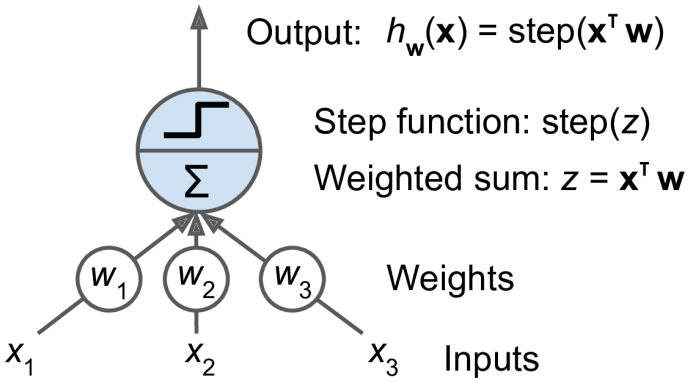


A5 Performance Measures



A6 Deep and Wide Supplementary Information

The diagram shows an example neuron in a neural network with a step function activation a. The activation function defines how the weighted sum of the input is transformed into an output from a node or nodes in a layer of the network.



Suppose we have 3 inputs. Then for each neuron, input-weight products are summed and then the sum is passed through a node's so-called activation function, to determine whether and to what extent that signal should progress further through the network to affect the output. This is represented by the following equations (note, the equation for a is dependent on the specific activation function chosen – step function in this example):

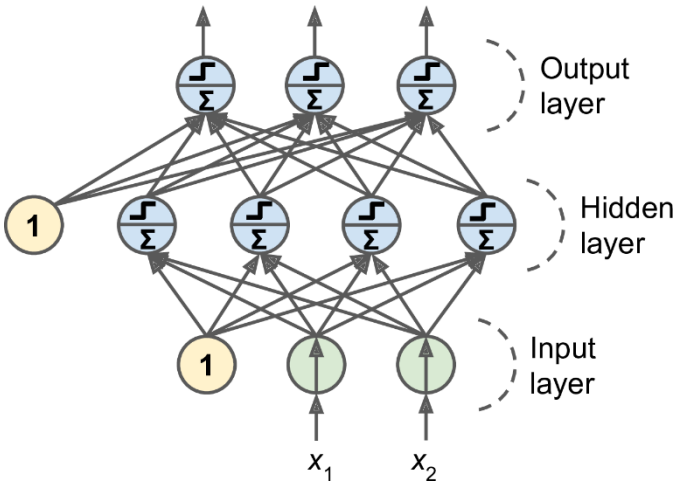
$$z = x_1 w_1 + x_2 w_2 + x_3 w_3 + b$$

$$a = step(z) = \begin{cases} 1 \text{ if } z > 0 \\ 0 \text{ if } z \leq 0 \end{cases}$$

A node layer consists of neuron-like switches that turn on or off as the input is fed through the neural network. Within each neuron, the weights and bias (from the equations above) are learned to minimise loss (MSE in this project). This helps identify which input is most helpful in the prediction problem.

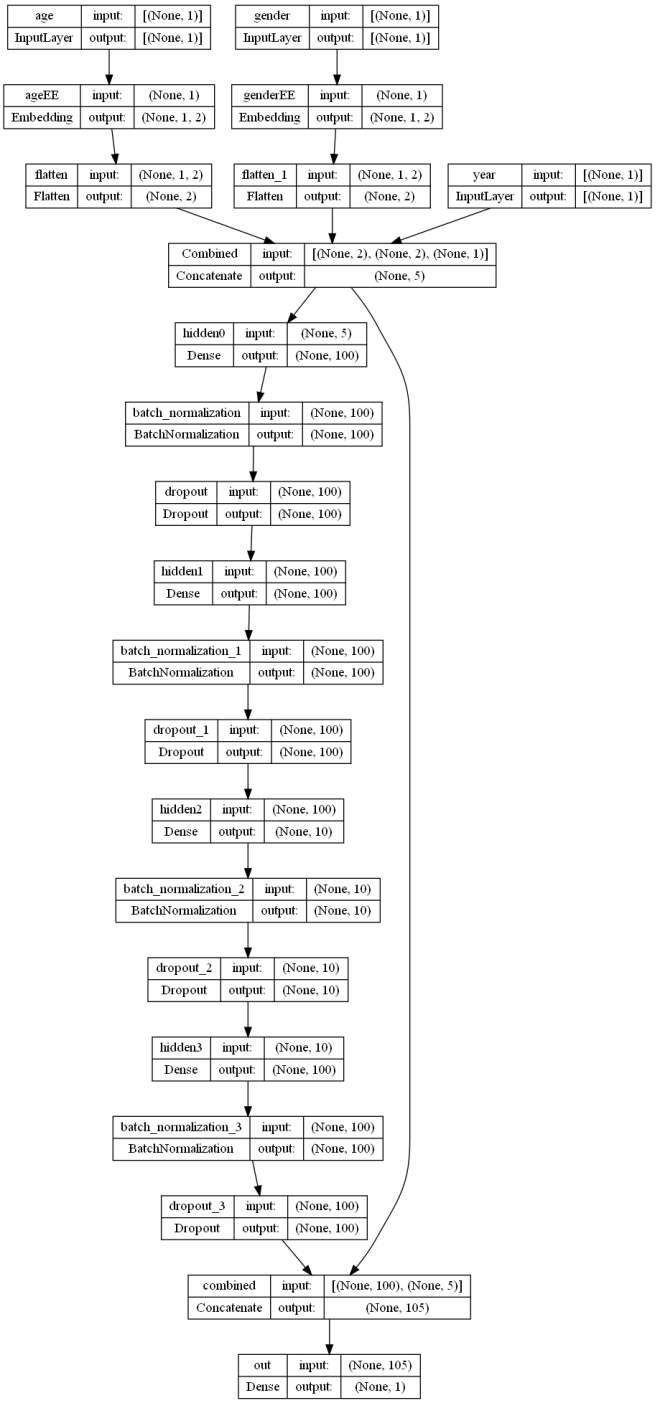
There are three types of node layers that make up the neural network:

- Input Layer: accept the data and pass it to the rest of the network.
- Hidden Layer: perform multiple functions at the same time such as data transformation, automatic feature creation, etc.
- Output Layer: concatenation of all layers to hold the result or the output of the problem

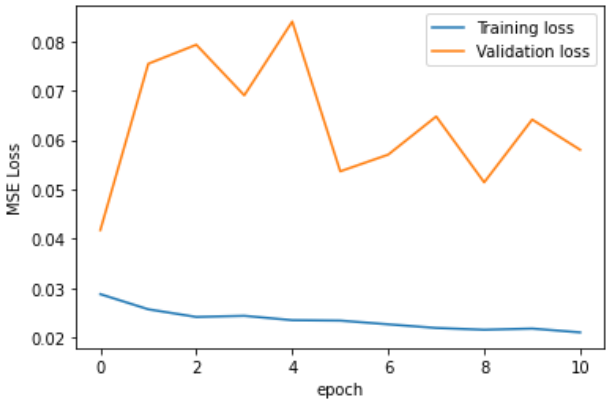


A7 Selected Deep and Wide Model

Model Diagram



Gradient Descent Algorithm



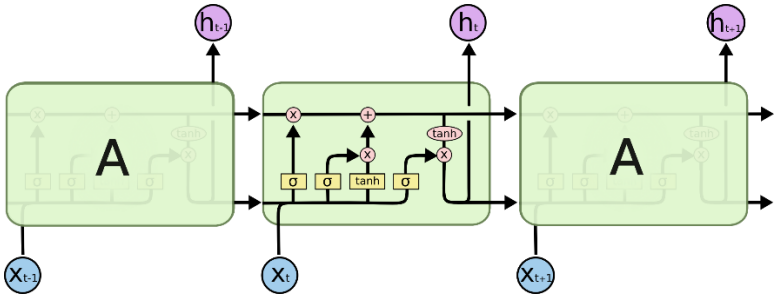
Model Summary

Layer (type)	Output Shape	Param #	Connected to
age (InputLayer)	[(None, 1)]	0	[]
gender (InputLayer)	[(None, 1)]	0	[]
ageEE (Embedding)	(None, 1, 2)	200	['age[0][0]']
genderEE (Embedding)	(None, 1, 2)	4	['gender[0][0]']
flatten (Flatten)	(None, 2)	0	['ageEE[0][0]']
flatten_1 (Flatten)	(None, 2)	0	['genderEE[0][0]']
year (InputLayer)	[(None, 1)]	0	[]
Combined (Concatenate)	(None, 5)	0	['flatten[0][0]', 'flatten_1[0][0]', 'year[0][0]']
hidden0 (Dense)	(None, 100)	600	['Combined[0][0]']
batch_normalization (BatchNormalization)	(None, 100)	400	['hidden0[0][0]']
dropout (Dropout)	(None, 100)	0	['batch_normalization[0][0]']
hidden1 (Dense)	(None, 100)	10100	['dropout[0][0]']
batch_normalization_1 (BatchNormalization)	(None, 100)	400	['hidden1[0][0]']
dropout_1 (Dropout)	(None, 100)	0	['batch_normalization_1[0][0]']
hidden2 (Dense)	(None, 10)	1010	['dropout_1[0][0]']
batch_normalization_2 (BatchNormalization)	(None, 10)	40	['hidden2[0][0]']
dropout_2 (Dropout)	(None, 10)	0	['batch_normalization_2[0][0]']
hidden3 (Dense)	(None, 100)	1100	['dropout_2[0][0]']
batch_normalization_3 (BatchNormalization)	(None, 100)	400	['hidden3[0][0]']
dropout_3 (Dropout)	(None, 100)	0	['batch_normalization_3[0][0]']
combined (Concatenate)	(None, 105)	0	['dropout_3[0][0]', 'Combined[0][0]']
out (Dense)	(None, 1)	106	['combined[0][0]']

Total params: 14,360
Trainable params: 13,740
Non-trainable params: 620

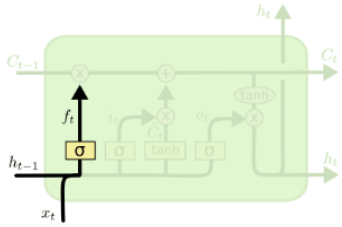
A8 LSTM Supplementary Information (Olah, 2015)

Information can be added or removed to the memory cell, regulated by structures called gates. This is composed out of a sigmoid layer that outputs numbers between 0 and 1 which describe how much of each component should be let through.



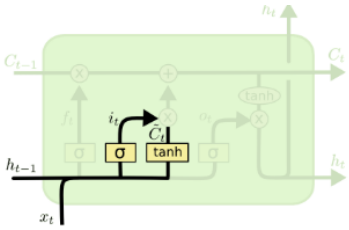
The process to decide what information to keep or thrown away is as follows:

- Information is passed through the forget gate layer which decides what is going to be thrown away from the cell state



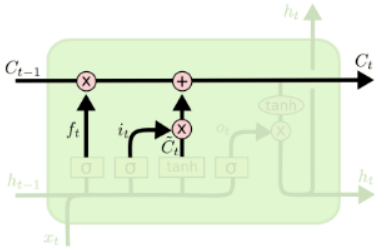
$$f_t = \sigma\left(W_f \cdot [h_{t-1}, x_t] \; + \; b_f\right)$$

- Input gate layer is used to decide which values will be updated
- Tanh layer creates a vector of new candidate values that could be added to the state



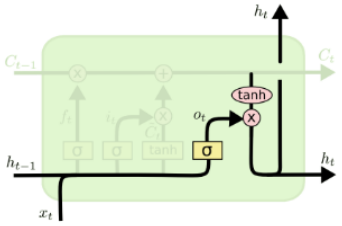
$$i_t = \sigma\left(W_i \cdot [h_{t-1}, x_t] \; + \; b_i\right)$$
$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] \; + \; b_C)$$

- Update the old cell state C_{t-1} into new cell state C_t



$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t$$

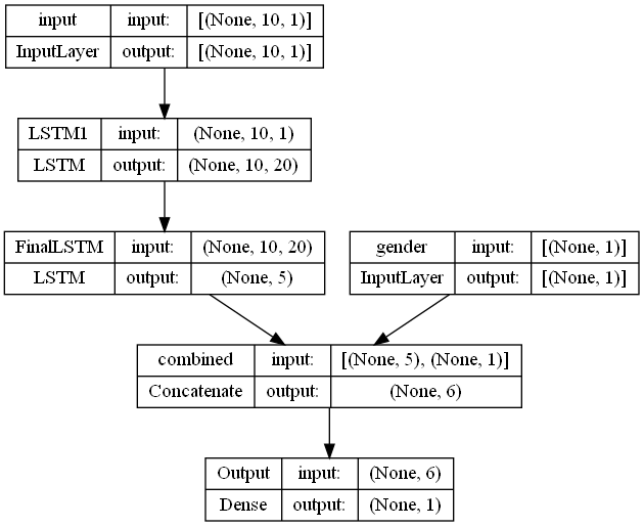
- Decide what is going to output



$$o_t = \sigma\left(W_o \; [h_{t-1}, x_t] \; + \; b_o\right)$$
$$h_t = o_t * \tanh\left(C_t\right)$$

A9 Selected LSTM Model

Model Diagram



Gradient Descent Algorithm

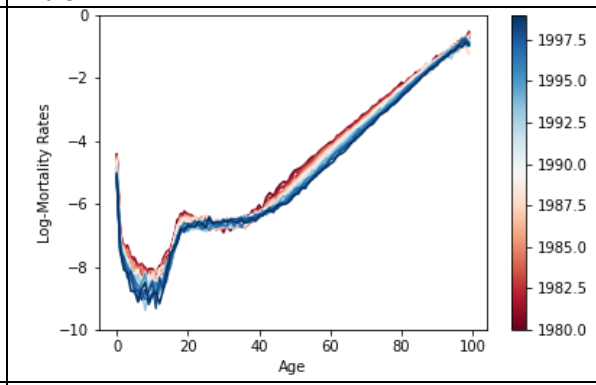
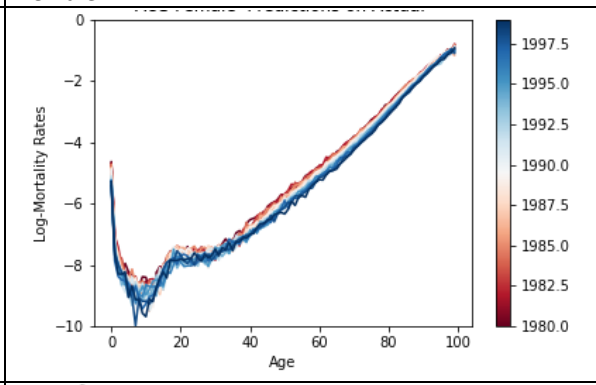
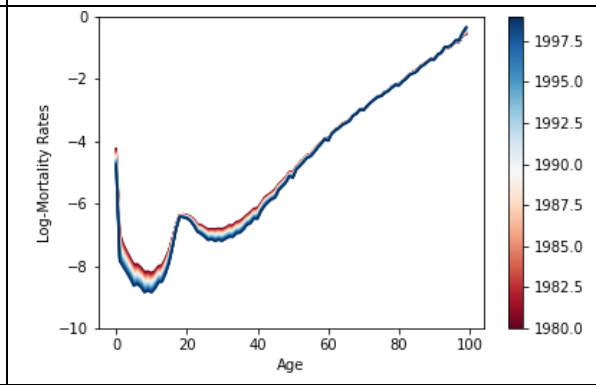
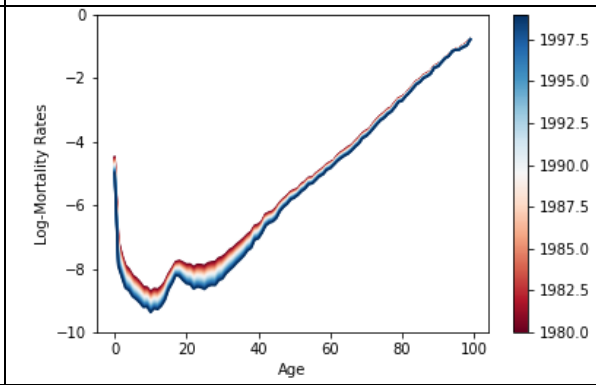
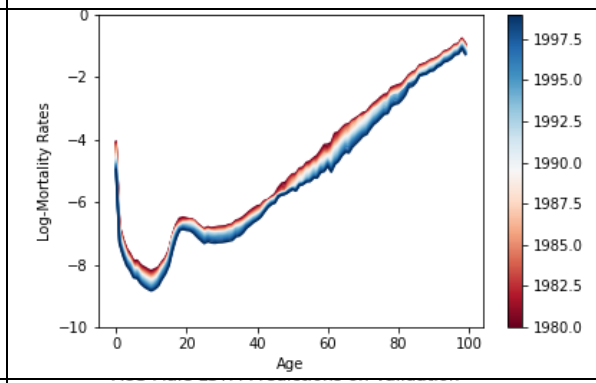
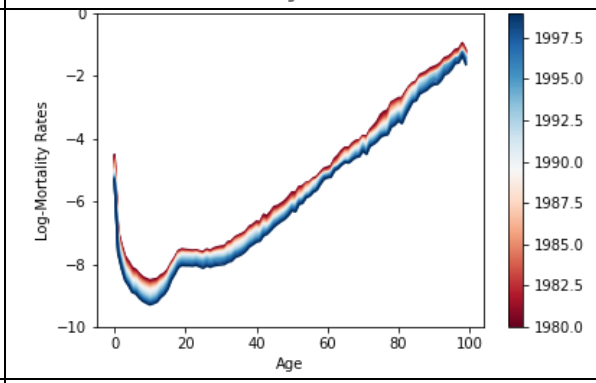
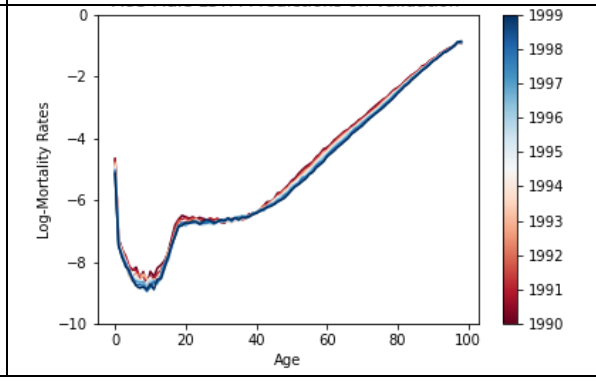
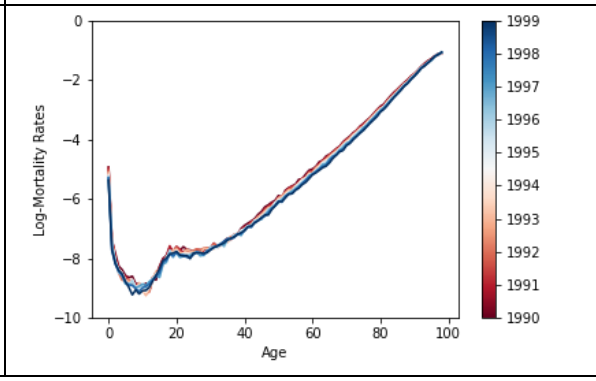
Model Summary

Layer (type)	Output Shape	Param #	Connected to
input (InputLayer)	[(None, 10, 1)]	0	[]
LSTM1 (LSTM)	(None, 10, 20)	1760	['input[0][0]']
FinalLSTM (LSTM)	(None, 5)	520	['LSTM1[0][0]']
gender (InputLayer)	[(None, 1)]	0	[]
combined (Concatenate)	(None, 6)	0	['FinalLSTM[0][0]', 'gender[0][0]']
Output (Dense)	(None, 1)	7	['combined[0][0]']

Total params: 2,287
Trainable params: 2,287
Non-trainable params: 0

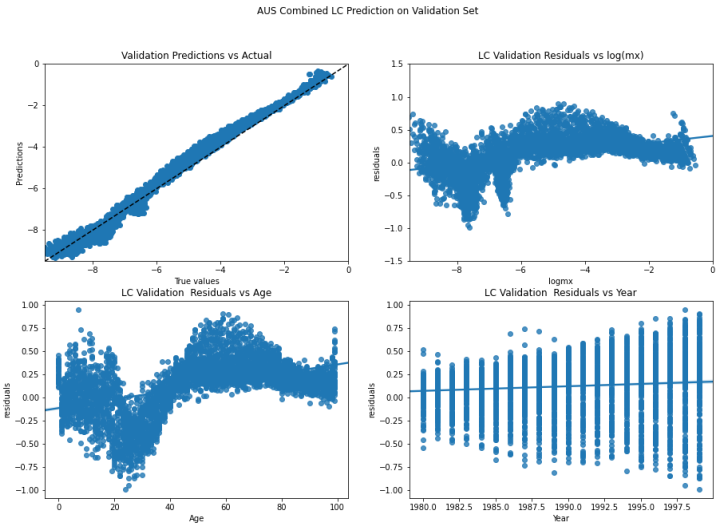
A10 Further Model Comparison

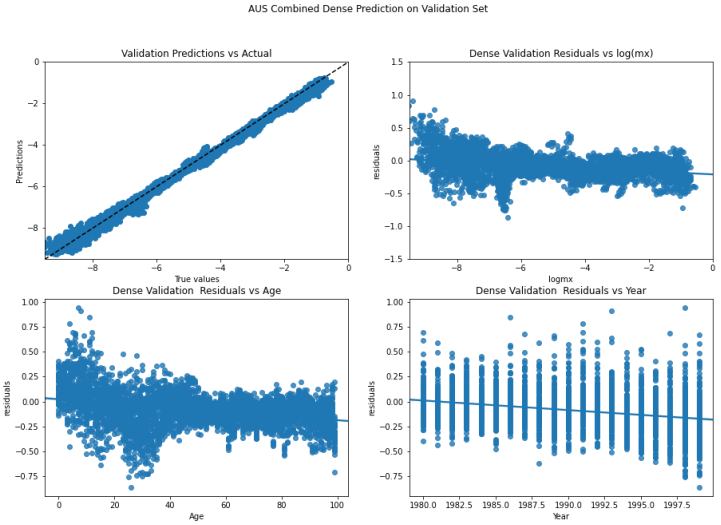
PREDICTION SHAPE

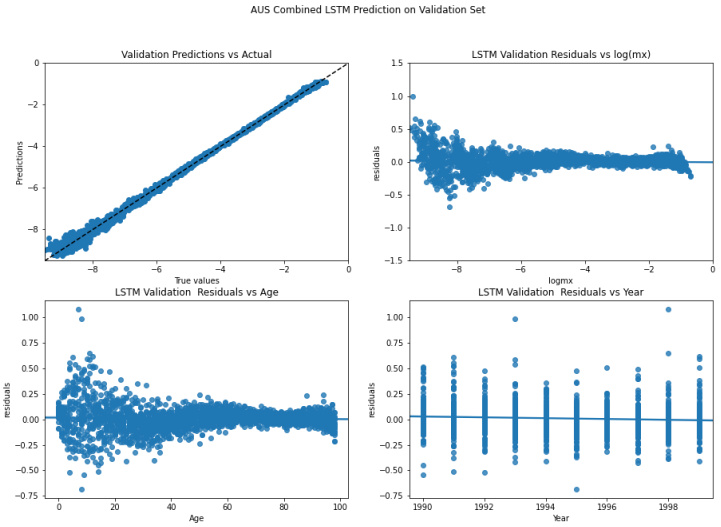
	Male	Female
Actual		
LC		
DW		
LSTM		

- All models reflect similar shape as the actual
- LSTM capture the volatility of the younger ages better
- LSTM also capture the more stable mortality rates between ages 20-40 better
- DW seems to capture the wide range of mortality of old age over time

COMBINED – VALIDATION



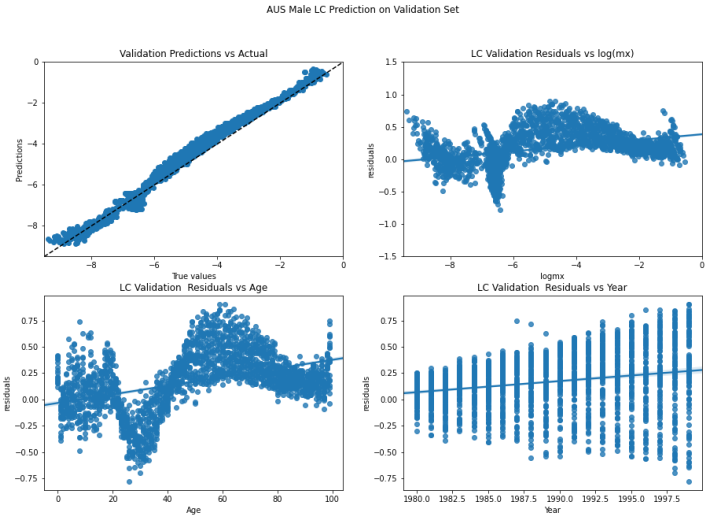


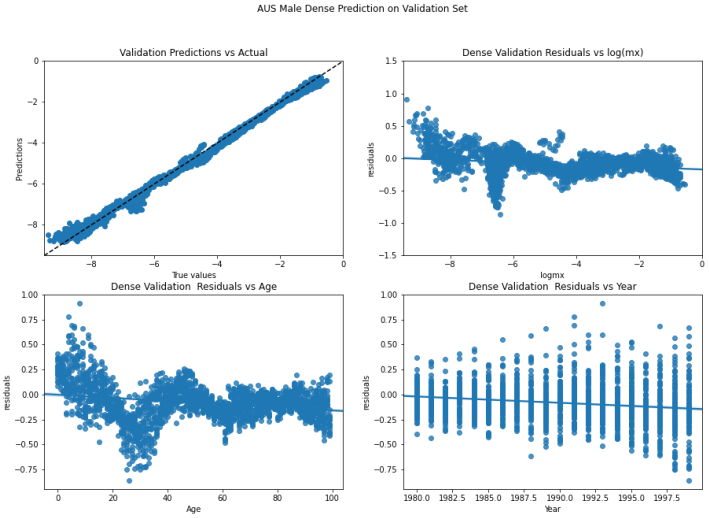


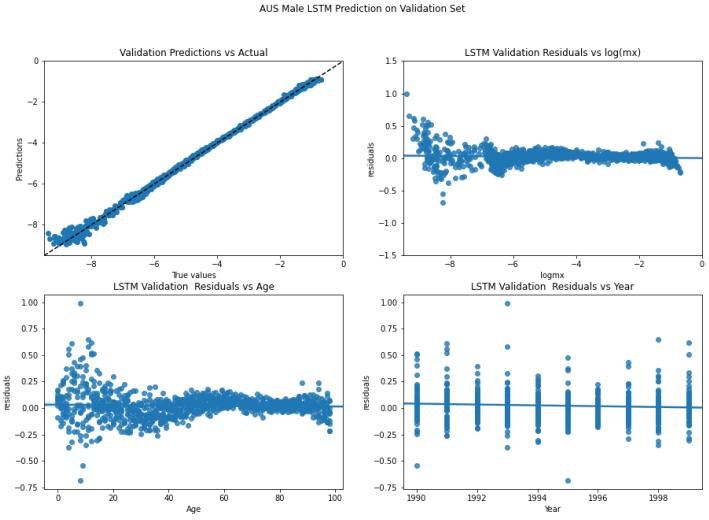
The following graphs show similar observations

- Top left: LSTM performs better as predictions lie more on the 45 degree line. This is shown by more of the residuals lying close to 0 in the other 3 graphs.
- Top right:
  - LC model shows increasing residuals as the observed log mortality increases – suggesting poorer performance for populations with higher mortality. The model also seems to underestimate lower mortality and overestimate higher mortality
  - DW and LSTM model shows less spread of the residuals. Although they both seem to perform poorer on lower mortality rates.
- Bottom left:
  - LC and DW model show an inability to fully capture age effects as shown by the wavy pattern of the LC model (overestimates young and old ages and underestimates middle age) and the downward sloping pattern of the DW model (underestimate old ages).
  - LSTM is more stable with residuals averaging 0 over ages. However, there is more spread in residuals in younger ages which reveals a slight inability to fully capture the volatility of mortality rates experienced by younger ages.
- Bottom right:
  - LC and DW model show an inability to fully capture period effects as shown by the respectively upward and downward sloping pattern in residuals. The LC model overestimates improvements in mortality over time whilst DW model underestimates period effects.

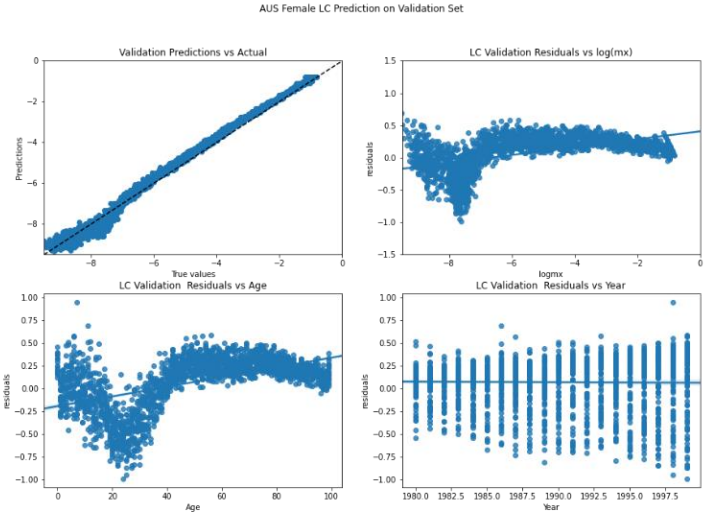
MALE – VALIDATION

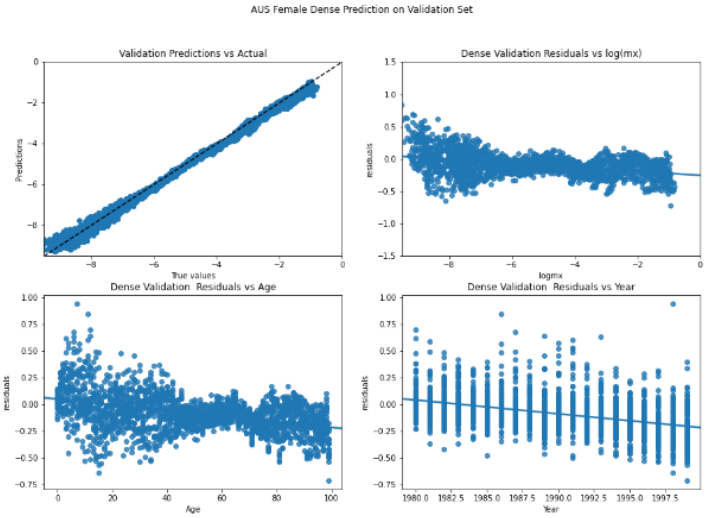


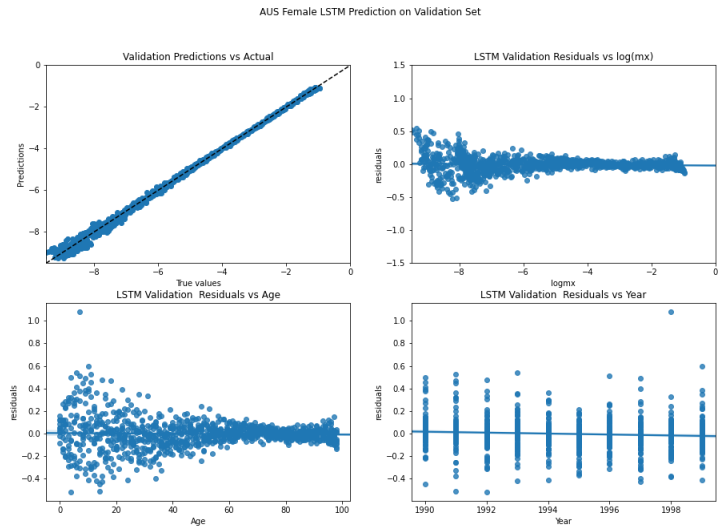




FEMALE - VALIDATION

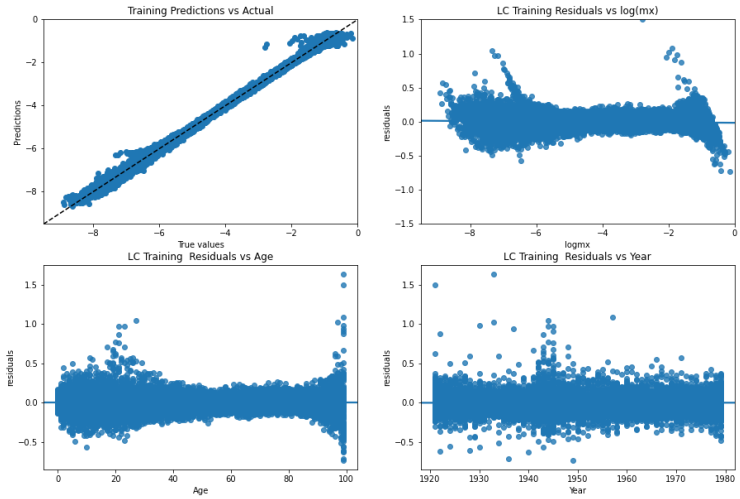




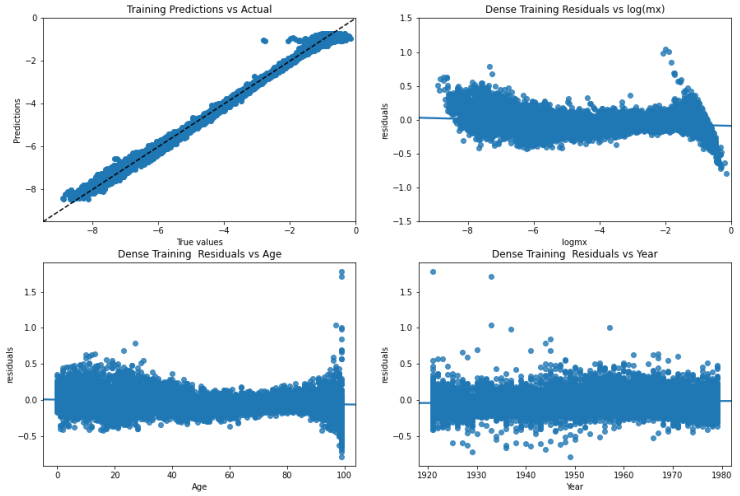


COMBINED – TRAINING

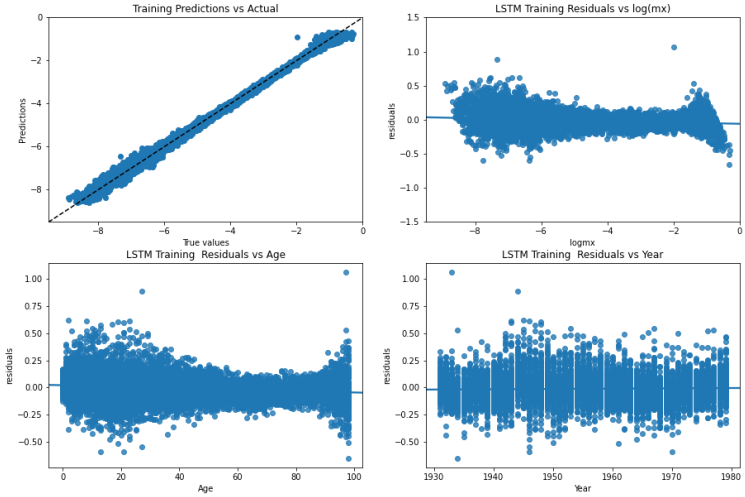
AUS Combined LC Prediction on Training Set



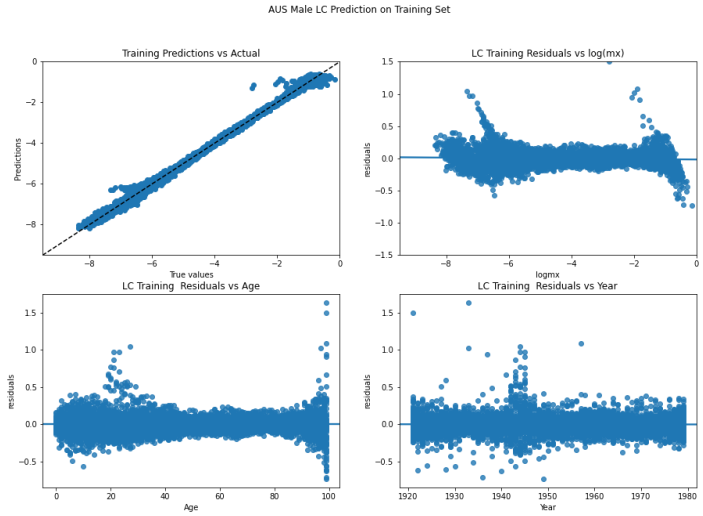
AUS Combined Dense Prediction on Training Set

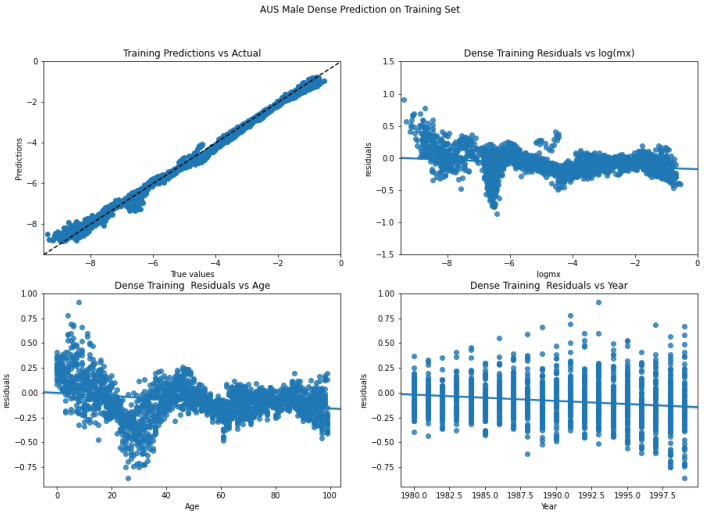


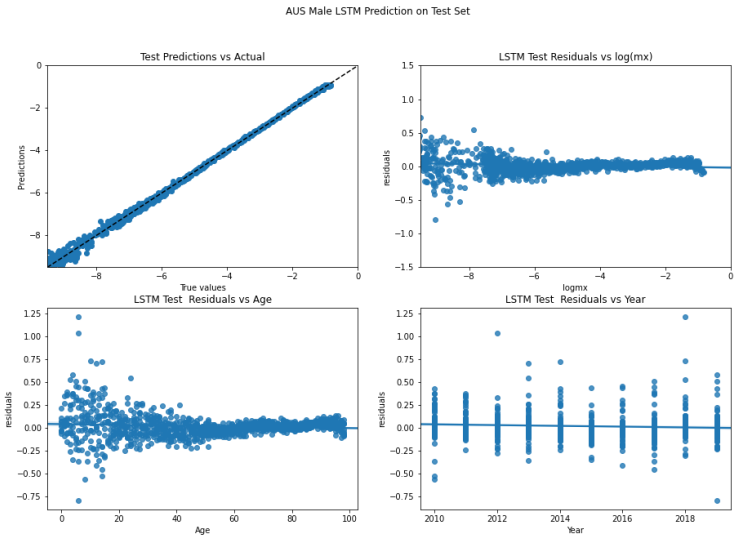
AUS Combined LSTM Prediction on Training Set



MALE – TRAINING

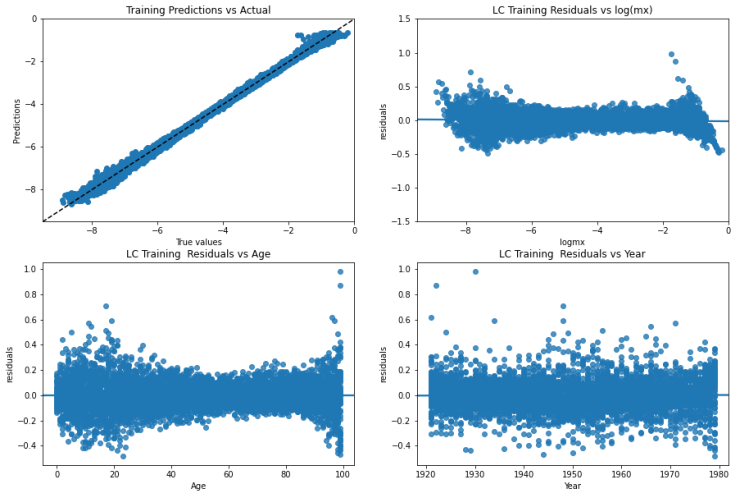




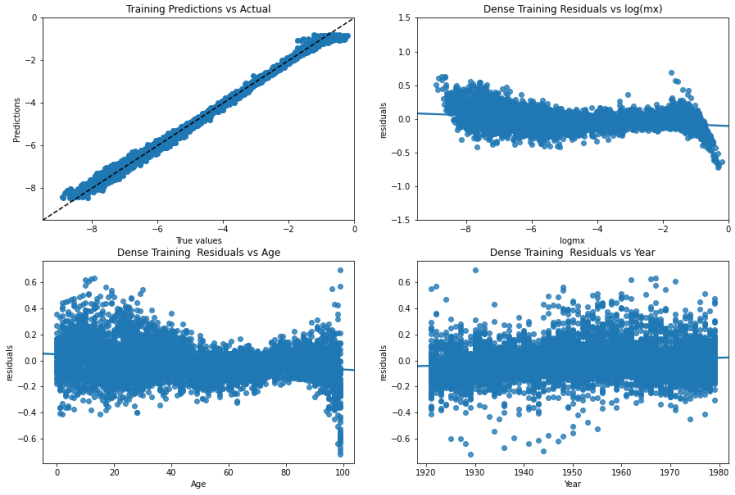


FEMALE – TRAINING

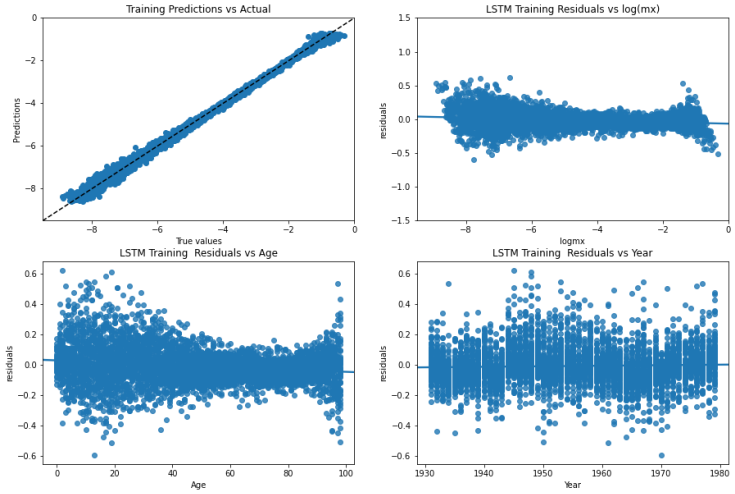
AUS Female LC Prediction on Training Set



AUS Female Dense Prediction on Training Set



AUS Female LSTM Prediction on Training Set



References

Schnürch, S. and Korn, R. (2022) "POINT AND INTERVAL FORECASTS OF DEATH RATES USING NEURAL NETWORKS," *ASTIN Bulletin*. Cambridge University Press, 52(1), pp. 333–360. doi: 10.1017/asb.2021.34. Available at: https://www.cambridge.org/core/journals/astin-bulletin-journal-of-the-iaa/article/point-and-interval-forecasts-of-death-rates-using-neural-networks/62F3A5B07463E5257E20B4FBD9115C6A

Richman, R. and Wüthrich, M. V. (2021) "A neural network extension of the Lee–Carter model to multiple populations," *Annals of Actuarial Science*. Cambridge University Press, 15(2), pp. 346–366. doi: 10.1017/S1748499519000071. Available at: https://www.cambridge.org/core/journals/annals-of-actuarial-science/article/neural-network-extension-of-the-leecartermodel-to-multiple-populations/19651C62C3976DCD73C79E57CF4A071C

Perla, Francesca and Richman, Ronald and Scognamiglio, Salvatore and Wuthrich, Mario V., Time-Series Forecasting of Mortality Rates using Deep Learning (May 6, 2020). Available at SSRN: https://ssrn.com/abstract=3595426 or http://dx.doi.org/10.2139/ssrn.3595426

Richman, Ronald and Wuthrich, Mario V., Lee and Carter go Machine Learning: Recurrent Neural Networks (2019). Available at SSRN: https://ssrn.com/abstract=3441030 or http://dx.doi.org/10.2139/ssrn.3441030

Olah, C (2015). Understanding LSTM Networks. Available at: http://colah.github.io/posts/2015-08-Understanding-LSTMs/

HMD (2019). Australia, Data Sources. Available at: https://www.mortality.org/File/GetDocument/hmd.v6/AUS/DOCS/ref.pdf

ABS (2022). Privacy at the ABS. Available at: https://www.abs.gov.au/about/legislation-and-policy/privacy/privacy-abs

Ioffe, Szegedy (2015). Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. Available at: https://arxiv.org/abs/1502.03167