Department of Actuarial Mathematics and Statistics

Full name: Gavin Yao Sheng Choong

Matriculation number: H00362702

Degree: Master of Science in Actuarial Management with Data Science (F7DM-AMD)

F71RA Machine Learning for Risk and Insurance: Project

Plagiarism declaration:

I confirm that I have read and understood: (a) the note on Plagiarism and collusion in the assignment handout; (b) the Heriot-Watt University regulations concerning plagiarism.

I confirm that the submitted work is my own and is in my own words.

I confirm that any source (aside from course notes and lecture material) from which I obtained information to complete this assignment is listed in the assignment. Any sources not listed in the assignment are listed here:

Apart from the lecturer, I discussed the assignment and shared ideas with the following people:

Signature: 

Date: 27th November 2023

Executive Summary

Machine learning is a subset of data science that is concerned with computer algorithms that can improve automatically by identifying patterns in data. Machine learning can be divided into three main areas: unsupervised, supervised and reinforcement learning. These machine learning techniques are powerful tools that have the potential to strongly influence the insurance industry, examples include:

1. Data manipulation and wrangling in order to obtain meaningful summary statistics regarding the average claim values on a yacht insurance policy amongst males and females.
2. Principal component analysis (PCA) to reduce the dimension of the features of the dataset, followed by linear regression in order to fit a model with the aim of predicting the claim amount of a new automobile insurance policy.
3. Deep neural networks which can be trained to eventually make predictions about the number of claims on a given policy.

These machine learning techniques can be used in the insurance industry in a wide variety of ways other than those mentioned above. With the potential to make predictions, machine learning techniques could improve competitiveness in the insurance industry through the enhancement of actuarial models and improvements in products and services offered.

As the use of machine learning techniques become increasingly widespread, however, it is crucial that fundamental guidelines and regulations such as the European Union's General Data Protection Regulation (GDPR) are implemented in order to achieve deployment of computer algorithms that are ethical and align with the three key principles of privacy, fairness and solidarity.

1 Introduction

The emergence and rapid development of big data has facilitated the adoption of data science which enables industries to continuously evolve, adapt and capitalize on new opportunities. For example, the implementation of machine learning methods in the insurance sector could potentially enhance classical actuarial models and create new products and services. Furthermore, insurance companies need to continuously adapt and evolve to cover new risks, in addition to those that typically fall within its remit, which may result from the combination of traditional data and big data.

Section 2 of this report explores the fundamental guidelines necessary for the ethical deployment of algorithms focusing on three key principles: privacy, fairness, and solidarity. This section also provides a concise summary of measures that can be implemented by insurance providers to uphold the three key principles. Additionally, this report delves into the different techniques that are commonly used to analyse data. The techniques considered are data manipulation and wrangling and unsupervised machine learning methods such as principal component analysis (PCA) and linear regression, and deep neural networks.

2 Regulations and Data Science Ethics

2.1 Ethical deployment of algorithms

With the rising popularity and rapid advancements in data science, it is more important than ever that fundamental guidelines are developed and implemented in order to ensure the ethical deployment of algorithms, focusing on three key principles of privacy, fairness, and solidarity in society. Addressing these key principles is vital in securing the rights of individuals and would promote equitable outcomes as these technologies become increasingly integrated in our daily lives.

In recent years, there has been a marked increase in consumer concerns regarding the collection and use of their personal data. This heightened awareness is largely attributable to privacy incidents, such as cases of data malpractice and data breaches which is expected to only become more frequent. Therefore, organizations need to implement guidelines in order to quell the privacy concerns of the public. Privacy protection could be implemented by adhering to the principle of data minimization which advocates for the minimal collection of essential data which would limit the impact in the case of a data breach. Additionally, organizations could adopt the concept of purpose limitation which involves clearly defining the objectives for data usage from the outset and ensuring that data usage is aligned with these established objectives, within reasonable expectations. The adoption of the concept of purpose limitation could enhance transparency and build consumer trust which would ultimately benefit the organization. Organizations could also implement more robust security measures such as data encryption technology to protect data from unauthorized users and prevent data breaches. The public's concern of privacy could be addressed through the implementation of and adherence to stringent privacy standards such as the European Union's General Data Protection Regulation (GDPR). However, it might prove costly for organizations to comply with.

The principle of fairness in the ethical deployment of algorithms can be achieved through the use of diverse and representative datasets collected from various sources to train algorithms with the aim of mitigating bias in the model. Regular independent audits of

algorithms and models would also contribute to the effort of ensuring fairness. In order to achieve solidarity in the deployment of algorithms, organizations could engage with a variety of stakeholders including affected communities to design algorithms that would benefit society especially marginalized communities.

2.2 Insurance Sector

The insurance industry is highly data driven and typically involves large quantities of sensitive personal data including risk factors (such as the age and gender), names, addresses, geolocation, and credit card information of the policyholder. Therefore, it is unsurprising that the industry is highly regulated. For example, models used in the insurance industry are required to meet transparency, fairness and solidarity requirements among policyholders. Furthermore, these models must be designed to assess risk based on relevant and non-discriminatory factors to ensure fairness and equality in underwriting and pricing of insurance products.

Insurance providers uphold the privacy of individuals through the implementation of appropriate and robust security measures such as data encryption technologies. Furthermore, insurers only collect data that is necessary for underwriting, claims processing and other processes with the consent of the individual, and the individuals are able to exercise their data protection rights to erase personal information and restrict or block processing of personal information.

The fairness in decision-making can be upheld by ethically designing algorithms by training models on diverse and representative datasets to mitigate the possibility of biased outcomes. This ensures that individuals are not being discriminated against by penalizing them through higher premiums based on factors that are not relevant to the risk being insured. Insurance providers should carry out periodic independent audits on the algorithms employed to ensure fair decision-making for all individuals. The behaviour and predictions of these algorithms should be explained to the stakeholders to improve transparency.

Insurance providers can promote solidarity by utilizing algorithms to identify new risks that may warrant the development of new insurance products or improvement of existing products to be more inclusive and better assist policyholders in mitigating risks.

3 Data Description and Preliminary Analysis

The data utilized in this report consists of three distinct datasets, each exemplifying a typical use case in the insurance sector.

3.1 Data Manipulation and Wrangling

A yacht insurance dataset was considered, and the structure of the data is described comprehensively in Figure 1.

```
tibble [1,340 × 6] (S3: tbl_df/tbl/data.frame)
 $ caseId                   : num [1:1340] 5 13 66 71 96 97 120 136 152 155 ...
 $ attorney                 : Factor w/ 2 levels "1","2": 1 2 2 1 2 1 1 1 2 2 ...
 $ claimantGender           : Factor w/ 2 levels "1","2": 1 2 1 1 1 2 1 2 2 1 ...
 $ containerDamage_insurance: Factor w/ 2 levels "1","2": 1 1 1 2 1 1 1 1 1 1 ...
 $ claimantAge              : num [1:1340] 50 28 5 32 30 35 19 34 61 NA ...
 $ loss                     : num [1:1340] 34.94 10.892 0.33 11.037 0.138 ...
```

Figure 1. The structure of the yacht insurance dataset.

This dataset consists of 1,340 claims made on yacht insurance policies and for each claim there are 6 variables whose summary statistics are shown in Figure 2.

```
     caseId      attorney claimantGender containerDamage_insurance  claimantAge         loss
 Min.   :    5   1:685    1   :586       1   :1270                 Min.   : 0.00   Min.   :      0.0
 1st Qu.: 8579   2:655    2   :742       2   :  22                 1st Qu.:19.00   1st Qu.:      0.6
 Median :17453            NA's: 12       NA's:  48                 Median :31.00   Median :      2.3
 Mean   :17213                                                     Mean   :32.53   Mean   :   3737.3
 3rd Qu.:25703                                                     3rd Qu.:43.00   3rd Qu.:      4.0
 Max.   :34253                                                     Max.   :95.00   Max.   :1000000.3
                                                                   NA's   :189
```

Figure 2. Summary statistics of the 6 columns in the dataset.

It is important to note for future sections that the variable `claimantGender` takes the value 1 if the claimant is a male and 2 if female and `NA` where the data was not available. The range, interquartile range and the 0.5 and 99.5 percentile of the `loss` variable, in Table 1, was calculated with the aim of identifying and replacing any outliers with `NA` to prevent them from affecting our calculations later.

Statistic	Value
Minimum value	0.005
Maximum value	1,000,000
Interquartile range	3.35575
0.5 percentile	0.0300
99.5 percentile	238.0207

Table 1. The range, interquartile range and the 0.5 and 99.5 percentile of `loss` variable.

The outliers were identified to be the values of `loss` which were below the 0.5 percentile and above 99.5 percentile. These outliers were replaced with `NA` and the entries related to these outlying `loss` values were removed since it is not needed in this report.

3.2 Principal Component Analysis (PCA) and Linear Regression

An automobile insurance dataset was analysed which contains the following 6 variables:

1. `carVal`: value of the automobile
2. `claimCount`: number of claims
3. `claimCost`: claim amount
4. `carAge`: age of automobile
5. `distance`: average travelling distance
6. `duration`: number of policy years

The correlation between variables are calculated and shown in Table 2 from which we can determine that the variables `carVal` and `carAge` are the most correlated with a correlation of −0.543511.

	carVal	claimCount	claimCost	carAge	distance	duration
carVal		0.0180	0.0098	−0.5435	−0.0049	−0.0017
claimCount	0.0180		0.4818	−0.0116	−0.0042	−0.0020
claimCost	0.0098	0.4818		0.0000	0.0051	−0.0027
carAge	−0.5435	−0.0116	0.0000		0.0050	0.0010
distance	−0.0049	−0.0042	0.0051	0.0050		−0.0025
duration	−0.0017	−0.0020	−0.0027	0.0010	−0.0025	

Table 2. Pearson's correlation between variables of the automobile insurance dataset.

3.3 Deep Neural Networks

The `freMTPL2freq` dataset used in this section which comprises of a French Motor Third-Part Liability (MTPL) Claims and the structure of this dataset is described in Figure 3.

```
'data.frame':   678013 obs. of  12 variables:
 $ IDpol     : num  1 3 5 10 11 13 15 17 18 21 ...
 $ ClaimNb   : num  1 1 1 1 1 1 1 1 1 1 ...
 $ Exposure  : num  0.1 0.77 0.75 0.09 0.84 0.52 0.45 0.27 0.71 0.15 ...
 $ Area      : Factor w/ 6 levels "A","B","C","D",..: 4 4 2 2 2 5 5 3 3 2 ...
 $ VehPower  : int  5 5 6 7 7 6 6 7 7 7 ...
 $ VehAge    : int  0 0 2 0 0 2 2 0 0 0 ...
 $ DrivAge   : int  55 55 52 46 46 38 38 33 33 41 ...
 $ BonusMalus: int  50 50 50 50 50 50 50 68 68 50 ...
 $ VehBrand  : Factor w/ 11 levels "B1","B10","B11",..: 4 4 4 4 4 4 4 4 4 4 ...
 $ VehGas    : Factor w/ 2 levels "Diesel","Regular": 2 2 1 1 1 2 2 1 1 1 ...
 $ Density   : int  1217 1217 54 76 76 3003 3003 137 137 60 ...
 $ Region    : Factor w/ 22 levels "R11","R21","R22",..: 18 18 3 15 15 8 8 20 20 12 ...
```

Figure 3. The structure of the `freMTPL2freq` dataset.

The variables `ClaimNb` and `Exposure` are defined as the number of claims on a given policy and the total exposure in yearly units respectively. Corrections were made to both `ClaimNb` and `Exposure` variables due to the belief of the presence of data errors. For the number of claims, there are only 9 policies observed to have more than 4 claims and these entries are corrected for by setting them equal to 4. From analysing the `Exposure` variable, it is observed that 1,224 exposures are larger than 1 year and these entries are corrected for by setting them equal to 1. Furthermore, the `IDpol` variable is excluded because it is only a unique identifier and not an explanatory variable.

4 Methods or Models

4.1 Data Manipulation and Wrangling

A Box-Cox transformation is a statistical technique used to transform non-normally distributed data into a shape that more closely approximates normality in order to reduce skewness in the data and stabilize the variance of the data. The one-parameter Box-Cox transformation used in this report is defined as follows:

$$y_i^{(\lambda)} = \begin{cases} \dfrac{y_i^{\lambda} - 1}{\lambda} & \text{if } \lambda \neq 0 \\ \log(y_i) & \text{if } \lambda = 0 \end{cases}$$

where the parameter λ is estimated using profile log-likelihood function is carried out in this report through the use of the `boxcox` function included in the `R` package `MASS`. This transformation is unsuitable for use with data that contains negative values. Therefore, it was essential to remove any negative values of the loss variable (which are the outliers since they are lower than the 0.5 percentile) before performing a Box-Cox transformation. The result of the `boxcox` function gives the optimal parameter $\lambda = 0$, therefore, the `loss` variable was transformed by the natural logarithm: $y_i^0 = \log(\texttt{loss})$.

4.2 Principal Component Analysis (PCA) and Linear Regression

4.2.1 Principal Component Analysis (PCA)

Principal component analysis (PCA) is an unsupervised machine learning method which aims at reducing the dimension of datasets by minimizing a square loss function. The PCA focuses only on continuous variables and assumes that these variables follow a Gaussian distribution. Hence, the `claimCount` variable is excluded when carrying out a PCA. Furthermore, since the goal of this section of the report is to fit a linear regression model to predict the `claimCost` of a policy, the `claimCost` variable is also excluded from the PCA.

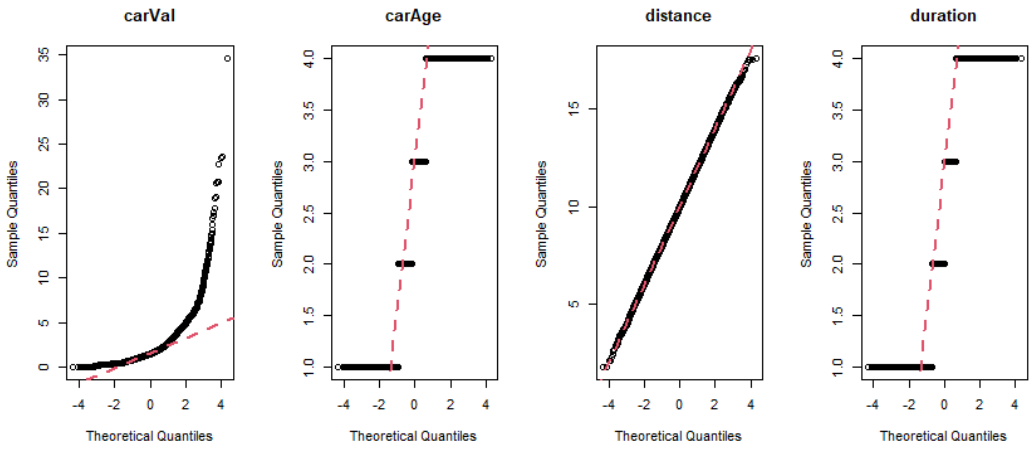

Figure 4. Normal Q-Q plots of the 4 variables considered for PCA.

From Figure 4, it can be observed that the `carVal` variable does not follow a Gaussian distribution and hence a natural logarithm transformation $\log(x+1)$ is applied to the `carVal` variable in order to reduce the skewness in the data as shown in Figure 5. A transformation $\log(x+1)$ was used instead of $\log(x)$ because `carVal` variable takes zero values. The PCA is carried out with `carVal.log`, `carAge`, `distance` and `duration` variables using the `princomp()` function in `R` where the data is scaled and centred by including the argument `cor = TRUE` in the function.

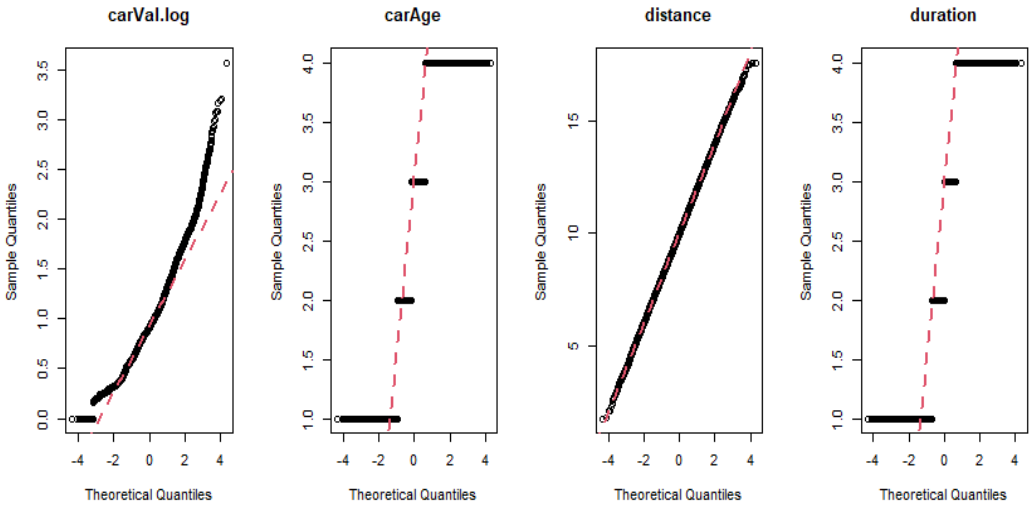

Figure 5. Normal Q-Q plots of the log transformed `loss` variable and the other 3 variables.

The scree plot in Figure 6 shows the proportion of explain variances by the individual principal components as well as the cumulative proportion of explained variances. It can be observed that the optimal model that retains 80% of the variation in the data includes the first 3 principal components.

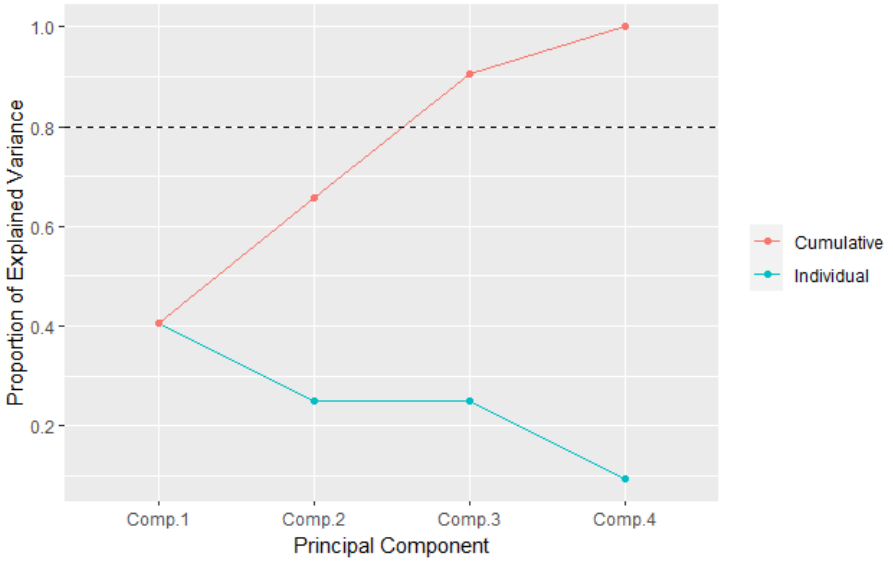

Figure 6. Scree plot of the PCA analysis showing the proportion of explained variances by individual principal components, and the cumulative proportion of explained variances.

4.2.2 Linear Regression

Linear regression is a supervised machine learning method used to predict the value of a response variable Y based on one or more explanatory variables X. This section of the report will make a prediction on the `claimCost` variable of an automobile insurance policy by first fitting a multiple linear regression model of the form:

$$Y_j = \beta_0 + \beta_1 X_{1,j} + \beta_2 X_{2,j} + \beta X_{3,j} + \epsilon_j$$

where the response variable Y_j represents the j^{th} observation of the `claimCost` variable and $X_{i,j}$ represents the j^{th} observation of the i^{th} principal component selected from the PCA above, for $i = 1, 2, 3$ and $j = 1, 2, \dots, 67856$. The β_i terms represent the coefficients whereas the β_0 term represents the intercept. Additionally, ϵ_i is the error variable that is assumed to be independent and identically distributed following a $N(0, \sigma^2)$ distribution.

The fitted regression model is then used to predict the `claimCost` of a new policy with the following specifications:

- `carVal` = 22
- `carAge` = 14.5
- `distance` = 25
- `duration` = 10

It important to note that the `carVal` variable needs to be transformed in the following manner: log (`carVal`+1).

4.3 Deep Neural Networks

A deep neural network consists of multiple hidden layers between the input and the output layer. It is common practice when training a neural network to split the data set into a learning dataset $\mathcal{D}$ and a test dataset $\mathcal{T}$. In this report, 90% of the dataset is

allocated as the learning dataset and the remaining 10% as the test dataset. Additionally, the learning dataset is partitioned such that 20% of the learning dataset is set apart as the validation set.

The combination of hyperparameters used for the neural network will be as follows:

- Number of hidden layers (depth): 3
- Number of neurons in each layer: 25, 20, 15
- Epochs: 1,000
- Batch size: 10,000
- Dimension of embedding layer: 2

The neural network applies a stochastic gradient descent (SGD) method to compute a parameter that minimizes an objective loss function. Therefore, continuous predictors require pre-processing using the Min-Max Scaler to ensure that all continuous predictors are on a similar scale between -1 and 1 since they are all of similar importance when performing SGD. The Min-Max scaler is a monotonic transformation defined as follows:

$$x^{(k)} \mapsto 2\frac{x^{(k)} - \min x^{(k)}}{\max x^{(k)} - \min x^{(k)}} - 1 \in [-1,1]$$

where the minimum and the maximum refer to the minimal and maximal value of the continuous predictor. However, the Min-Max scaler will not be useful if outliers are present in the continuous predictors.

The categorical feature components, `VehBrand` and `Region` are embedded into embedding layers, where each observation of these categorical features will be represented by a 2-dimensional vector. Additionally, a log transformation of the variable `Exposure`, log (`Exposure`), will be incorporated as a non-trainable offset. The data structure is then defined for the `Design` matrix, which contains all continuous predictors, and categorical features as well as the non-trainable offset log (`Exposure`).

The main architecture of the neural network is then constructed with 3 hidden layers between the input and output layer. The input layer for the `freMTPL2freq` dataset are the risk characteristics of the policyholders and their cars. The hidden layers employ a hyperbolic tangent 'tanh' activation function whereas the output layer employs an exponential activation function which is consistent with the use of the Poisson deviance loss as the objective loss function for the neural network since the use of an exponential activation function ensures that the outputs take only positive values which is appropriate for count data such as `claimNb` which is the response variable. The Nadam optimizer is the optimization algorithm used in this section since it outperforms other gradient-based optimization algorithms in terms of correctness.

The fitting process of the deep neural network is carried out using the learning dataset $\mathcal{D}$ with 20% of the dataset used as the validation set. The learning and testing results are recorded, and the deviance loss for the neural network on both the learning dataset $\mathcal{D}$ and the test dataset $\mathcal{T}$ are computed.

5 Results and Discussion

5.1 Data Manipulation and Wrangling

When applying the Box-Cox transformation on the `loss` variable, the distribution of the data more closely resembles a normal distribution as can be seen in Figure 7.

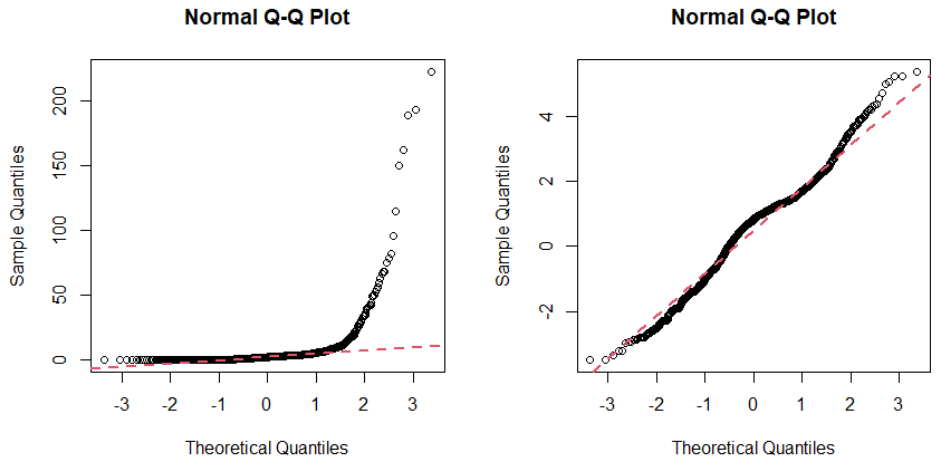

Figure 7. Normal Q-Q plots of the `loss` variable (left) and the Box-Cox transformed `loss` variable (right)

Further analysis reveals that the mean claim values amongst males is 5.66 which is higher than that amongst females which is 4.43. Categorizing the claimants according to their ages as in Table 3, it can be observed that the number of claimants is generally lower at higher ages except for the age category from 43 to 72 which has the second highest number of claimants.

Age Category of Claimants	(Strictly) under 25	From 26 to 35	From 36 to 42	From 43 to 72	72 or above
Count	421	224	172	299	25

Table 3. Number of claimants at each age category.

5.2 Principal Component Analysis (PCA) and Linear Regression

The biplots in Figure 8, shows biplots between the first 3 principal components which accounts for 40.6%, 25.1% and 24.9% of the variance in the data respectively. The grey points on the biplots represent the standardized data points which seem to be densely clustered around the origin which could indicate a high degree of similarity across the observations of the dataset.

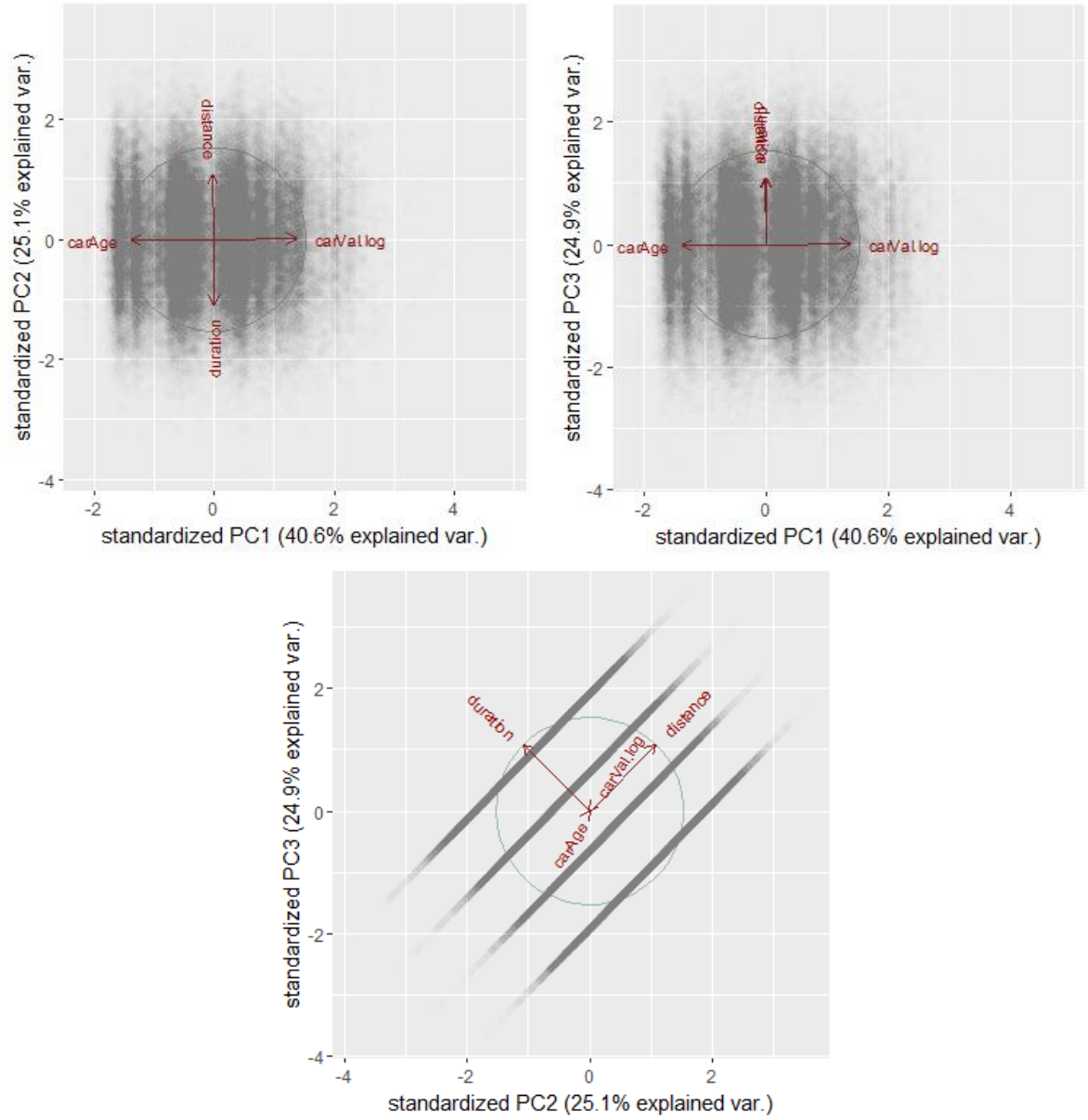

Figure 8. Biplots between the first 3 principal components chosen from PCA.

The `claimCost` of the new policy with the specifications as previously discussed in Section 4.2.2 is predicted using the fitted regression model to be 102.8502.

5.3 Deep Neural Networks

Figure 9 shows the training loss and validation loss, represented by the blue and green line respectively, over the number of epochs during the training of the deep neural network. The training loss and validation loss is defined as the error of the model on the learning dataset and the validation set respectively. It can be observed that the training

loss decreases steeply at the beginning and appears to converge towards the validation loss at higher numbers of epoch.

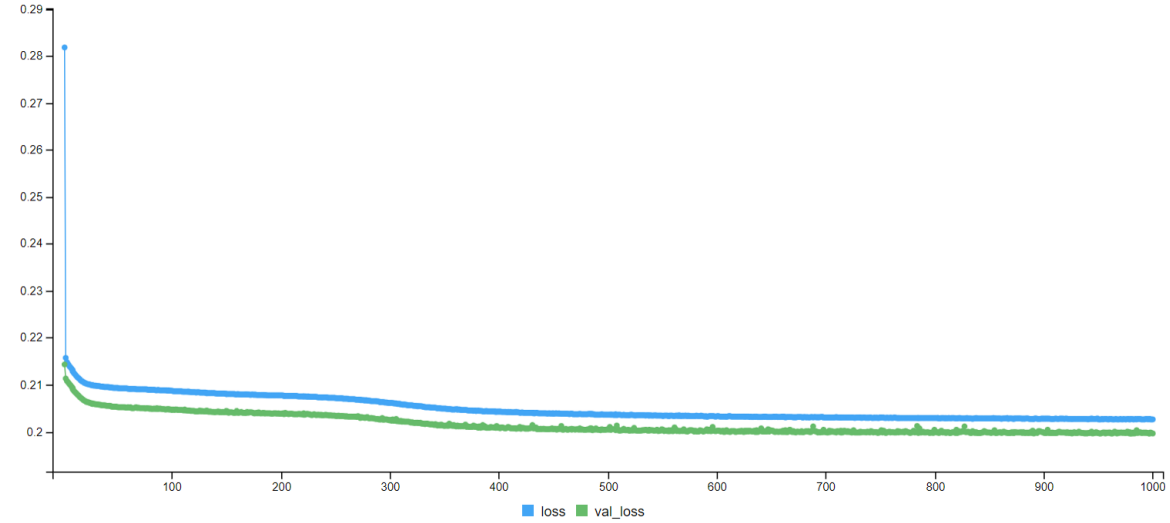

Figure 9. Training loss and validation loss over the number of epochs during the training of the deep neural network.

The deviance loss for the deep neural network on the learning and test dataset was 0.3056 and 0.3004 respectively. The deviance loss values on both datasets are close, therefore, it is reasonable to conclude that the trained deep neural network generalizes well to unseen data.

If the deviance loss to be utilized in the training process of the deep neural network described in section 4.3 is to be of the Poisson-Inverse Gaussian distribution instead of the Poisson distribution, the deviance loss function needs to be modified. Given the probability mass function of the Poisson-Inverse Gaussian distribution:

$$\mathbb{P}(k|\,\mu,\sigma) = \left(\frac{2\alpha}{\pi}\right)^{0.5} \frac{\mu^k e^{\frac{1}{\sigma}} K_\alpha}{(\alpha\sigma)^k k!}$$

where $\alpha^2 = \frac{1}{\sigma^2} + \frac{2\mu}{\sigma}$, for $k = 0, 1, 2, \ldots$ is the number of claims, where $\mu > 0$ and $\sigma > 0$, $K_\lambda(t) = \frac{1}{2}\int_0^\infty x^{\lambda-1} \exp\left\{-\frac{1}{2}t\left(x + \frac{1}{x}\right)\right\} dx$ is the modified Bessel function of the third kind.

For a single observation k_i, consider the log-likelihood function, $l(\mu, \sigma|k_i)$:

$$\begin{aligned} l(\mu,\sigma|k_i) &= \log\left(\mathbb{P}_i(k_i|\,\mu,\sigma)\right) \\ &= \log\left(\left(\frac{2\alpha}{\pi}\right)^{0.5} \frac{\mu^{k_i} e^{\frac{1}{\sigma}} K_\alpha}{(\alpha\sigma)^{k_i} k_i!}\right) \\ &= 0.5\log\left(\frac{2\alpha}{\pi}\right) + k_i \log\mu + \frac{1}{\sigma} + \log(K_\alpha) - k_i\log(\alpha\sigma) - \log(k_i!) \end{aligned}$$

The deviance loss, D, of the Poisson-Inverse Gaussian distribution would be:

$$D = 2\sum\left(l(\mu_i = k_i, \sigma_i|k_i) - l(\hat{\mu}_i, \hat{\sigma}_i|k_i)\right)$$

The parameter α would need to be calculated for each observation i and the modified Bessel function of the third kind K_α would need to be evaluated for each α. Additionally,

$\hat{\mu}_i$ and $\hat{\sigma}_i$ need to be estimated. The stochastic gradient descent (SGD) would also need to compute $\hat{\mu}_i$ and $\hat{\sigma}_i$. Therefore, the output layer should also output two values, one for $\hat{\mu}_i$ and $\hat{\sigma}_i$.

6 Conclusions

This report has shown that machine learning techniques can be very useful in the insurance industry as it can be used to obtain meaningful summary statistics, reduce the dimensions of high dimensional datasets and make predictions on the claim amount and number of claims on a policy. However, regulations and guidelines need to be implemented to ensure the ethical deployment of algorithms based on the principles of privacy, fairness, and solidarity.

Bibliography

Hodge, R. (2019) 2019 Data breaches hall of shame: These were the biggest data breaches of the year. CNET. Accessed from: https://www.cnet.com/news/privacy/2019-data-breach-hall-of-shame-these-were-the-biggest-data-breaches-of-the-year/

Information Commissioner's Office. UK GDPR guidance and resources. Accessed from: https://ico.org.uk/for-organisations/uk-gdpr-guidance-and-resources/

Institute and Faculty of Actuariese (IFoA). (2021). Ethical and professional guidance on Data Science: A Guide for Members

Maurya, M., Yadav, N. (2023). A Comparative Analysis of Gradient-Based Optimization Methods for Machine Learning Problems. In: Yadav, A., Gupta, G., Rana, P., Kim, J.H. (eds) Proceedings on International Conference on Data Analytics and Computing. ICDAC 2022. Lecture Notes on Data Engineering and Communications Technologies, vol 175. Springer, Singapore. https://doi.org/10.1007/978-981-99-3432-4_7

Noll, A., Salzmann, R., Wuthrich, M. (2020). Case Study: French Motor Third-Party Liability Claims. SSRN Electronic Journal. doi: 10.2139/ssrn.3164764.

Petrostan, A. (2023) Online Privacy in the United Kingdom (UK) – Statistics & Facts. Accessed from: https://www.statista.com/topics/7342/online-privacy-in-the-uk/#editorsPicks

Quach, S., Thaichon, P., Martin, K.D. *et al.* Digital technologies: tensions in privacy and data. *J. of the Acad. Mark. Sci.* **50**, 1299–1323 (2022). https://doi.org/10.1007/s11747-022-00845-y

Shlens, J. (2014). A Tutorial on Principal Component Analysis. *Educational* 51.

Appendix

Table of Contents for R functions

Complete List of Code

The following is the R code that was used in the making of this report.

```
#Removing all objects from memory.
rm(list=ls())
```

```r
#Loading required packages.
library(tidyverse)
library(keras)
library(reticulate)
library(ggplot2)
library(cluster)
library(ClusterR)
library(readxl)
library(magrittr)
library(dplyr)

##################################Part2#########################################
###Question (a)
#Importing data into R as a data frame.
mydata.2 = read_xlsx("yacht_insurance.xlsx", col_names = T)

#Create factors for the following columns.
mydata.2$attorney = factor(mydata.2$attorney)
mydata.2$claimantGender = factor(mydata.2$claimantGender)
mydata.2$containerDamage_insurance = factor(mydata.2$containerDamage_insurance)

#Structure and summary statistics for each column.
str(mydata.2)
summary(mydata.2)

###Question (b)
#Calculating the range and interquartile range of the loss column.
range(mydata.2$loss); IQR(mydata.2$loss)
#Calculating the 0.5 and 99.5 percentile of the loss column.
q = quantile(mydata.2$loss, c(0.005, 0.995)); q
#Replacing the outliers, which lie below the 0.5 percentile and above the 99.5 percentile,
#with NA.
mydata.2$loss[mydata.2$loss < q[1] | mydata.2$loss > q[2]] = NA

###Question (c)
#Deleting entries related to outlying loss values indicated by NA.
mydata.2.filtered = filter(mydata.2, !is.na(mydata.2$loss))

###Question (d)
#Loading MASS in order to use the boxcox function
library(MASS)

#Boxcox function to estimate optimal lambda using maximum likelihood estimation
bc = boxcox(mydata.2.filtered$loss ~ 1, plotit = F)
#Choosing optimal value of lambda
lambda = bc$x[which.max(bc$y)]; lambda
mydata.2.filtered = mutate(mydata.2.filtered, BC.Loss = log(mydata.2.filtered$loss))

#Unloading the MASS package since it clashes with dplyr when using select function
detach("package:MASS", unload=TRUE)

#Plotting normal Q-Q plot to show the Box-Cox transformed variable closely #resembles a nor
#mal distribution
par(mfrow = c(1,2))
qqnorm(mydata.2.filtered$loss); qqline(mydata.2.filtered$loss, col = 2, lty = 2, lwd = 2)
qqnorm(mydata.2.filtered$BC.Loss); qqline(mydata.2.filtered$BC.Loss, col = 2, lty = 2, lwd
= 2)

###Question (e)
#Grouping the data by gender and calculating the mean loss for each gender
mydata.2.filtered %>% group_by(claimantGender) %>% summarise(avg = mean(loss))
```

```r
###Question (f)
#Creating a categorical variable based on age of claimants
mydata.2.filtered$age.category = cut(mydata.2.filtered$claimantAge,
                                     breaks = c(0, 25, 35, 42, 72, Inf),
                                     labels = c("(strictly) under 25", " from 26 to 35",
                                                "from 36-42", "from 43-72", "72 or above"),
                                     right = F
)
#Count of the number of claimants in each category
mydata.2.filtered$age.category = factor(mydata.2.filtered$age.category)
summary(mydata.2.filtered$age.category)

##################################Part3#########################################
#Importing data into R as a data frame.
mydata.3 = read_xlsx("automobile.xlsx", col_names = T)

###Question (a)
#Calculating the correlation between the columns using only complete cases.
corr.matrix = cor(mydata.3, use = "complete.obs")
#Setting the diagonal of the matrix to NA since it always equals to 1.
diag(corr.matrix) = NA; round(corr.matrix, 4)
#Selecting the columns that has the highest correlation .
most.corr = which(abs(corr.matrix) == max(abs(corr.matrix), na.rm = T), arr.ind = T)
#Displaying the variables that are the most correlated with the correlation value.
most.corr; corr.matrix[1,4]

####Question (b)
#Excluding claimCount and claimCost
mydata.3.filtered = dplyr::select(mydata.3, !claimCount & !claimCost)

##Normal Q-Q plots of untransformed data excluding claimCount and claimCost
temp.1 = colnames(mydata.3.filtered)
par(mfrow = c(1,length(temp.1)))
for(i in 1:length(temp.1)){
  var.1 = mydata.3.filtered %>% dplyr::select(temp.1[i]) %>% pull
  qqnorm(var.1, main = temp.1[i]); qqline(var.1, col = 2, lty = 2, lwd = 2)
}

#Log transformation of data
mydata.3.trans = mydata.3.filtered %>% mutate(carVal.log = log(carVal + 1))
#Removing untransformed carVal column
mydata.3.trans = dplyr::select(mydata.3.trans, !carVal)
#Rearranging columns of the data frame
mydata.3.trans = mydata.3.trans[,c(4,1:3)]

#Normal Q-Q plots of transformed data
temp.2 = colnames(mydata.3.trans)
par(mfrow = c(1,length(temp.2)))
for(i in 1:length(temp.2)){
  var.2 = mydata.3.trans %>% dplyr::select(temp.2[i]) %>% pull
  qqnorm(var.2, main = temp.2[i]); qqline(var.2, col = 2, lty = 2, lwd = 2)
}

#PCA with cor = TRUE to scale and center the data
mydata.3.pca = princomp(mydata.3.trans, cor = TRUE)
summary(mydata.3.pca)
#Calclating the cumulative variance explained by the PC
cum.var = cumsum(mydata.3.pca$sdev^2); total.var = sum(mydata.3.pca$sdev^2)
#Selecting least number of PC that retain 80% of variation in data
PC.retained = which((cum.var / total.var) >= 0.8)[1]; PC.retained

###Question (c)
##Skree Plot
#Preparing the data frame containing proportion of variance explained by individual PC and
```

```r
#cumulative variance
temp.3 = data.frame(mydata.3.pca$sdev^2 / total.var, cum.prop.var = cum.var/total.var)

#Plotting the skree plot
ggplot(data = temp.3) +
  geom_path(mapping = aes(x = row.names(temp.3),y = temp.3[,1], color = "Individual"), grou
p = 1) +
  geom_point(mapping = aes(x = row.names(temp.3),y = temp.3[,1], color = "Individual")) +
  geom_path(mapping = aes(x = row.names(temp.3), y = temp.3[,2], color = "Cumulative"),grou
p = 1) +
  geom_point(mapping = aes(x = row.names(temp.3), y = temp.3[,2], color = "Cumulative")) +
  #Adding a y-intercept at 0.8
  geom_hline(aes(yintercept = 0.8), linetype = 2) +
  labs(x = "Principal Component", y = "Proportion of Explained Variance") +
  guides() +
  theme(legend.position = "right", legend.title = element_blank()) +
  #Specifying scale of the y axis
  scale_y_continuous(breaks  = seq(from = 0, to = 1, by = 0.2))

##Biplot
library(ggbiplot)
#Plotting a biplot between the 3 PC selected, alpha = 0.004 chosen to improve readability
#of plot
ggbiplot(mydata.3.pca, choices = c(1,2), alpha = 0.004, circle = T, ellipse = T, varname.si
ze = 3)
ggbiplot(mydata.3.pca, choices = c(1,3), alpha = 0.004, circle = T, ellipse = T, varname.si
ze = 3)
ggbiplot(mydata.3.pca, choices = c(2,3), alpha = 0.004, circle = T, ellipse = T, varname.si
ze = 3)



###Question (d)
#Preparing data frame consisting of the claimCost and scores of the first 3 PC
model.df = data.frame(mydata.3$claimCost, mydata.3.pca$scores[,-4])
colnames(model.df)[1] = "claimCost"
#Fitting the linear model with the first 3 PC
model.3 = lm(claimCost ~ Comp.1 + Comp.2 + Comp.3, data = model.df); summary(model.3)

###Question (e)
#New set of features
carVal.new = 22; carAge.new = 14.5; distance.new = 25; duration.new = 10
#Transforming data
carVal.new.log = log(carVal.new + 1)
#Preparing vector of data to be used for prediction
new.obs = c(duration.new, distance.new, carVal.new.log, carAge.new)
#Scaling and centering the data to be used for prediction
new.obs.standardized = (new.obs - mydata.3.pca$center)/mydata.3.pca$scale

#Preparing a matrix containing the loadings of the first 3 PC
PC = matrix(nrow = 4, ncol = 3)
for(i in 1:3){
  PC[,i] = mydata.3.pca$loadings[,i]
}

#Matrix multiplication between data used for prediction and loadings
PC.new.obs = as.data.frame(new.obs.standardized %*% PC)
colnames(PC.new.obs) = colnames(model.df[,-1])
#Predicting the claimCost of policy with the specifications of the new set of #features
prediction = predict(model.3, newdata = PC.new.obs); prediction
```

```r
################################Part4#########################################
#Importing data into R as a data frame.
mydata.4 = read.csv("freMTPL2freq.csv")

#Corrections made to the dataset
mydata.4$ClaimNb = pmin(mydata.4$ClaimNb, 4)
mydata.4$Exposure = pmin(mydata.4$Exposure, 1)

##Create factors for columns that has a character data type.
for(i in seq_along(mydata.4)){
  if(is.character(mydata.4[[i]])){
    mydata.4[[i]] <- factor(mydata.4[[i]])
  }
}

###Question (a)
str(mydata.4)

#Author of function: Dr George Tzougas
#Function obtained from Week 9 Lecture Slides Part VI B
#Min-Max scaler used to normalize continuous predictors between -1 and 1
MM_scaling = function(data){
  2 * (data - min(data))/(max(data) - min(data)) - 1
}

#Min-Max Scaling of continuous predictors
mydata.4.NN = data.frame(ClaimNb = mydata.4$ClaimNb)
mydata.4.NN$Area = MM_scaling(as.integer(mydata.4$Area))
mydata.4.NN$VehPower = MM_scaling(as.numeric(mydata.4$VehPower))
mydata.4.NN$VehAge = MM_scaling(as.numeric(mydata.4$VehAge))
mydata.4.NN$DrivAge = MM_scaling(mydata.4$DrivAge)
mydata.4.NN$BonusMalus = MM_scaling(mydata.4$BonusMalus)
mydata.4.NN$VehGas = MM_scaling(as.integer(mydata.4$VehGas))
mydata.4.NN$Density = MM_scaling(mydata.4$Density)

#Learning Testing Split
#Learning Sample Index
learn.idx = sample(1:nrow(mydata.4), round(0.9 * nrow(mydata.4)), replace = F)
#Learning Testing Split
learn = mydata.4[learn.idx,] ; test = mydata.4[-learn.idx,]
learn.NN = mydata.4.NN[learn.idx,] ; test.NN = mydata.4.NN[-learn.idx,]

#Embedding Layers
library(tensorflow)
#Number of vehicle brands and region
Br_ndistinct = length(unique(learn$VehBrand))
Re_ndistinct = length(unique(learn$Region))

#Setting up input layer for categorical features
VehBrand = layer_input(shape = c(1), dtype = 'int32', name = 'VehBrand')
Region = layer_input(shape = (1), dtype = 'int32', name = 'Region')

#Dimension of embedding layer for VehBrand and Region
qEmb = 2

#Creating embedding layer for VehBrand and Region
BrEmb = VehBrand %>%
  layer_embedding(input_dim = Br_ndistinct, output_dim = qEmb,
                  input_length = 1, name = "BrEMB") %>%
  layer_flatten(name = "Br_flat")
ReEmb = Region %>%
  layer_embedding(input_dim = Re_ndistinct, output_dim = qEmb,
                  input_length = 1, name = 'ReEmb') %>%
  layer_flatten(name = 'Re_flat')
```

```r
###Question (b)
#Set design matrix for continuous variables
Design.learn = as.matrix(learn.NN[, -1]); Design.test = as.matrix(test.NN[, -1])

#Set matrices for categorical variables
Br.learn = as.matrix(as.integer(learn$VehBrand)) - 1; Br.test = as.matrix(as.integer(test$V
ehBrand)) - 1
Re.learn = as.matrix(as.integer(learn$Region)) - 1; Re.test = as.matrix(as.integer(test$Reg
ion)) - 1

#Non-trainable offset
Exp.learn = as.matrix(learn$Exposure); Exp.test = as.matrix(test$Exposure)
Exp.log.learn = log(Exp.learn); Exp.log.test = log(Exp.test)

#Matrix for response variable
Y.learn = as.matrix(learn.NN$ClaimNb); Y.test = as.matrix(test.NN$ClaimNb)

##Neural Network
#Setting hyperparameters
#Number of neurons in each layer
q1 = 25; q2 = 10; q3 = 15
#Number of epochs and number of samples per gradient upgrade
epochs = 1000; batchsize = 10000

#Setting up Input layer for continuous features
Design = layer_input(shape = ncol(learn.NN) - 1, dtype = 'float32', name = "Design")

#Setting up Input layer for log(Exposure) as offset
Exp.log = layer_input(shape = c(1), dtype = 'float32', name = 'Exp.log')
Exp = layer_input(shape = c(1), dtype = 'float32', name = 'Exp')

#Main architecture with 3 hidden layers
Network = list(Design, BrEmb, ReEmb) %>% layer_concatenate(name = 'concate') %>%

  #1st hidden layer
  layer_dense(units = q1, activation = 'tanh', name = 'hidden1') %>%

  #2nd hidden layer
  layer_dense(units = q2, activation = 'tanh', name = 'hidden2') %>%

  #3rd hidden layer
  layer_dense(units = q3, activation = 'tanh', name = 'hidden3') %>%

  #output layer (w/ one neuron only)
  layer_dense(units = 1, activation = 'linear', name = 'Network')


#Output layer to combine main architecture and offset layer
Response = list(Network, Exp.log) %>%

  #Adding exposure and the last neuron
  layer_add() %>%

  #Give the response
  layer_dense(units = 1,
              activation = 'exponential',
              name = 'Response',
              trainable = FALSE,
              weights = list(array(1, dim = c(1,1)), array(0,dim = c(1))))
```

```r
#Assembling the model
model.4 = keras_model(inputs = c(Design, VehBrand, Region, Exp.log),
                      outputs = c(Response))
summary(model.4)



#Configuring the model
model.4 %>% compile(
  #Poisson deviance loss
  loss = 'poisson',
  #Nadam optimizer
  optimizer = 'nadam'
)
#Model fitting by running gradient descent method to minimize obj. function
{
  t1 = proc.time()

  fit = model.4 %>% fit(
    list(Design.learn, Br.learn, Re.learn, Exp.log.learn), #Predictors
    Y.learn, #Response

    verbose = 1,

    epoch = epochs,

    batch_size = batchsize,

    validation_split = 0.2 #20% of learning dataset as validation set
  )
  print(proc.time() - t1)
}



#Predicted value of claim numbers
learn$nn0 = as.vector(model.4 %>% predict(list(Design.learn, Br.learn, Re.learn, Exp.log.le
arn)))
test$nn0 = as.vector(model.4 %>% predict(list(Design.test, Br.test, Re.test, Exp.log.test))
)
#Author of function: Dr George Tzougas
#Function obtained from Week 8 Slides II Lecture Notes Part VI A
#Used to calculate the Poisson deviance loss
dev.loss = function(y, mu, density.func){
  logL.tilde = log(density.func(y, y))
  logL.hat = log(density.func(y, mu))
  2 * mean(logL.tilde - logL.hat)
}


#Deviance loss for the deep neural network on learning dataset
dev.loss(y = learn$ClaimNb, mu = learn$nn0, density.func = dpois)
#Deviance loss for the deep neural network on teset dataset
dev.loss(y = test$ClaimNb, mu = test$nn0, density.func = dpois)
```