

Proyecto de Consultoría para Promutual,
compañía de seguros para negligencia médica.

Cálculo de reserva por incurridos no avisados (IBNR) usando Maching Learning.

ado en las lineas

4 de diciembre de 2023

Elizabeth Suárez Sánchez.
esuares@unal.edu.co

Índice general

1. Resumen ejecutivo.	3
1.1. CRISP-DM Entendimiento del negocio.	3
1.2. Introducción.	4
1.3. Objetivos del Proyecto:	5
1.3.1. Objetivos especificos	5
2. Metodología.	7
2.1. Chain Ladder.	8
2.2. Regresión Lineal Chain Ladder.	12
2.3. Lasso y Ridge.	15
2.3.1. Ridge.	15
2.3.2. Lasso.	17
2.4. Redes Neuronales.	18
3. Datos.	21
3.1. CRISP DM- Preparación de la data.	22
3.2. CRISP DM- Entendimiento de la data.	23
4. Entrenamiento del modelo.	28
4.1. CRISP DM- Modelado.	28
4.2. Partición de datos.	28
4.3. Validación cruzada.	30
5. Implementacion.	31
5.1. CRISP DM- Evaluado.	31
5.1.1. Red Neuronal.	31
5.2. CRISP DM- Desarrollo.	33
5.2.1. Modelos Determinístico.	33

5.2.2. Modelos de regresión.	34
5.2.3. Red Neuronal.	37
6. Resultados.	42
6.1. Regresión lineal y Chain Ladder.	42
6.2. Validación.	42
6.3. Test.	43
6.4. Red Neuronal.	43
6.5. Comparativos gnerales.	43
7. Riesgos y Limitaciones.	46
8. Conclusiones	47
Referencias	48

Capítulo 1

Resumen ejecutivo.

1.1. CRISP-DM Entendimiento del negocio.



Como empresa de consultoría, nos comprometemos a ofrecer a las empresas la más destacada asesoría, destacándonos como impulsores de la vanguardia y siendo plenamente conscientes de las responsabilidades que las aseguradoras tienen con sus inversionistas.

En este contexto, el presente estudio tiene como objetivo llevar a cabo una evaluación exhaustiva de las reservas de IBNR mediante diversas metodologías. Además de emplear una metodología determinística ampliamente reconocida, como el método Chain Ladder, hemos explorado otras metodologías que incorporan avances en Machine Learning, tales como Lasso, Ridge y redes neuronales.

La aplicación de un análisis integral utilizando diversas herramientas innovadoras proporciona a las aseguradoras la confianza de perfeccionar las predicciones sobre sus reservas, mostrándose precavidos pero sin descuidar sus inversiones. Este enfoque holístico busca dotar a las empresas del sector de seguros con la certeza de estar utilizando las estrategias más avanzadas y eficaces en la gestión de sus recursos financieros. Estamos comprometidos

a ser su socio estratégico en el camino hacia la excelencia en la toma de decisiones y la optimización de sus operaciones.

1.2. Introducción.

Promutual es una aseguradora especializada en el sector médico, proporcionando cobertura a médicos contra posibles demandas por negligencia médica. Estas demandas pueden implicar sumas considerables que afectan directamente a los hospitales, representando un riesgo significativo para la sostenibilidad de un centro de salud que no cuenta con una cobertura adecuada para este tipo de eventualidades. Aunque la probabilidad de tales eventos es baja, su impacto es grande cuando ocurren.

Dado que Promutual opera como aseguradora para estos casos, es crucial que pueda anticipar posibles eventualidades a lo largo del tiempo. Esto le permite estar preparada para responder a las demandas y, al mismo tiempo, ser rentable para sus inversionistas.

Si Promutual no realiza un cálculo preciso de la reserva necesaria para el próximo año y un hospital asegurado recibe una demanda por negligencia de gran envergadura, la empresa no solo pone en riesgo su propia estabilidad, sino también la del hospital y de los inversionistas. Por esta razón, las aseguradoras están sujetas a la atención de las entidades regulatorias que velan por la protección tanto del inversionista como del asegurado.

Supongamos, por otro lado, que Promutual calcula de manera precisa sus reservas y responde adecuadamente a las demandas anuales. En este escenario, la aseguradora evitará problemas con el hospital, pero podría enfrentar desafíos con los inversionistas, quienes podrían considerar no invertir en una aseguradora que no tiene liquidez. Así, el cálculo de la reserva no solo debe cubrir los siniestros, sino también ser rentable.

Un cálculo inteligente del IBNR permite a Promutual tomar decisiones estratégicas que le permitan competir eficazmente en el mercado de seguros.

El problema de los seguros para negligencia médica son incluso de gran actualidad, pues hasta hace unos meses Netflix sacó un documental llamado “Cuiden a Maya”, este documental relata la historia de la familia Kowalski quienes sufrieron por años el lidiar con diagnósticos equivocados sobre la enfermedad de su hija Maya, lo que además de resultar en

traumas para la familia y para su hija, ocasionó un gran deterioro mental en la madre quien decidió suicidarse, pues el hospital Johns Hopkins tenía retenida a su hija bajo una acusación falsa de posible maltrato por parte de la madre. Cuatro años después de lo ocurrido el juez de Florida determinó que hubo negligencia médica y la suma de la demanda fue de 261 millones de dólares (infobae, 2023).

Más allá de los hechos legales, 261 millones de dólares es una suma que perfectamente puede fracturar a un hospital, a no ser que estén cubiertos por un seguro por negligencia médica .

En resumen, el cálculo preciso del IBNR es fundamental para el funcionamiento integral de una aseguradora y, por extensión, para el mercado completo de bienes y servicios.



1.3. Objetivos del Proyecto:

El objetivo principal de este proyecto consiste en proporcionarle a Promutual un proceso sistemático en el software R que permita calcular su reserva de IBNR de cuatro formas diferentes, las cuales son: Regresión lineal, lasso , ridge y por redes neurales, todas estas formas se ponen a prueba con la metodología tradicional de chain Ladder.

1.3.1. Objetivos específicos

- Obtener la data adecuada con la cual se va a entrenar los modelos.

- Analizar la data y sacar conclusiones.
- Aplicar el modelo determinístico chain Ladder sobre la data de Promutual.
- Aplicar el modelo de regresión lineal chain Ladder sobre la base de Promutual y compararla con la de chain Ladder determinístico.
- Hacer una validación cruzada que recorra todas las posibles combinaciones sobre mi base de entrenamiento y determinar qué modelo entre lasso, ridge y regresión tiene el mejor MAPE (Error Porcentual Medio).
- Implementar redes Neuronales para encontrar el IBNR y compararlo con los demás modelos, entre ellos el determinístico.
- Establecer cuál modelo se aplica mejor a las necesidades de Promutual.

En capítulos posteriores, profundizaremos un poco más en la información sobre los datos.



Capítulo 2

Metodología.

El IBNR por sus siglas en inglés(Incurred but not reported) hace referencia a la reserva de siniestros que ya ocurrieron a la fecha del cálculo pero aún no son reportado. Actuarialmente se busca estimar daños potenciales que pudiera afrontar la aseguradora (Promutual) pero que a la fecha de estimar una reserva se desconoce, es por eso que la base de partida para este cálculo es el histórico de incurridos (o siniestros ocurridos) que se han ido pagando en el tiempo desde la fecha de ocurrencia o accidente ([Facecolda, 2023](#)) .

A partir del pasado, con la base histórica de pagos se busca estimar el futuro, por eso lo primero que se hace es transformar esa tabla de años de accidente, años de desarrollo y pagos (incurridos) en triángulos, que es la presentación base para aplicar la metodología tradicional de chain Ladder y para el caso de este proyecto también será la base para las metodologías de regresión lineal, lasso y ridge.

Antes de adentrarnos en cada una de las metodologías, es fundamental definir la estructura de un triángulo de incurridos. Estos triángulos tienen en el eje horizontal los periodos de desarrollo y en el vertical los años desde los cuales tenemos información o datos de incurridos.

A triangle: 10 × 10 of type dbl

	0	1	2	3	4	5	6	7	8	9
1988	43393	39560	50398	46630	44007	30000	27171	27395	27034	26804
1989	23460	54478	50633	46938	34432	30406	30408	29794	28014	NA
1990	44843	40405	38517	29661	23559	23269	22813	22594	NA	NA
1991	40715	44679	35510	33872	32713	30871	28734	NA	NA	NA
1992	55743	44309	33884	29723	30954	27665	NA	NA	NA	NA
1993	49446	30760	27841	37179	35687	NA	NA	NA	NA	NA
1994	33465	32438	38458	37845	NA	NA	NA	NA	NA	NA
1995	32837	32449	34650	NA	NA	NA	NA	NA	NA	NA
1996	34711	31935	NA	NA	NA	NA	NA	NA	NA	NA
1997	34107	NA	NA	NA	NA	NA	NA	NA	NA	NA

En el caso de Promutual, contamos con un triángulo que abarca los años de accidente desde 1988 hasta 1997, y los años de desarrollo se extienden desde el año del accidente hasta 10 años en el futuro. Estos años de desarrollo se presentan de manera horizontal. Por ejemplo, la primera fila representa los pagos realizados desde 1988 hasta 1997 (hasta la fecha actual). Esto significa que todos los valores en la diagonal son los pagos más recientes, es decir, aquellos con fecha hasta 1997.

Siguiendo lo mencionado anteriormente, el triángulo superior representa todos los pagos conocidos, es decir, anteriores a la fecha actual (1997), mientras que los valores del triángulo inferior representan aquellos pagos que aún no se han realizado, es decir, que ocurrieron después de 1997 y se extienden hasta 2006.

En la práctica, los triángulos inferiores no son conocidos (es lo que se espera estimar), pero como deseamos construir un modelo de machine learning, necesitamos ambos triángulos, el superior y el inferior (la parte con NA del aimagen superior) , con los cuales podamos entrenar nuestro modelo y ponerlo a prueba.

2.1. Chain Ladder.

El problema que se intenta resolver con Chain Ladder consiste en calcular de forma precisa cuánto puede costar un siniestro y cuánto puede tardar en pagarse ese siniestro.




Figura 2.1: Imagen modificada de citio web Facecolda.

El proceso por el cual pasa una aseguradora cuando ocurre un siniestro y se le es reportado es el siguiente:

Para el caso de un seguro por negligencia médica, el tiempo que transcurre entre que ocurre el evento y es reportado a la aseguradora puede ser bastante largo, ya que todo el aspecto legal entra a jugar un papel importante. En la gran mayoría de los casos las demandas pueden tardar varios años y es necesario que la aseguradora pueda proveer el posible costo de la demanda durante este tiempo y que en caso de que el juez determiné que hubo negligencia, la aseguradora tenga los recursos (Facecolda, 2023) .

Una vez entendido el problema, se parte del triángulo acumulado.


Triángulo acumulado: Es el triángulo superior en donde para cada año de desarrollo se suman los pagos de los años anteriores, esto quiere decir que en la última columna de año de desarrollo se obtiene el monto total de lo pagado por los accidentes ocurridos en el año particular de accidente durante toda su vida de desarrollo. Para el caso de Promutual el triángulo acumulado es el siguiente:

 A triangle: 10 × 10 of type dbl

	0	1	2	3	4	5	6	7	8	9
1988	43393	82953	133351	179981	223988	253988	281159	308554	335588	362392
1989	23460	77938	128571	175509	209941	240347	270755	300549	328563	NA
1990	44843	85248	123765	153426	176985	200254	223067	245661	NA	NA
1991	40715	85394	120904	154776	187489	218360	247094	NA	NA	NA
1992	55743	100052	133936	163659	194613	222278	NA	NA	NA	NA
1993	49446	80206	108047	145226	180913	NA	NA	NA	NA	NA
1994	33465	65903	104361	142206	NA	NA	NA	NA	NA	NA
1995	32837	65286	99936	NA	NA	NA	NA	NA	NA	NA
1996	34711	66646	NA	NA	NA	NA	NA	NA	NA	NA
1997	34107	NA	NA	NA	NA	NA	NA	NA	NA	NA

Una vez se tiene el triángulo acumulado, se calculan los factores de desarrollo.

Para el cálculo de los factores de desarrollo se va a colocar una coordenada a cada celda del triángulo donde j recorre los años de desarrollo y i el año de accidente, de esta forma la celda $C(i, j) = c(1, 2)$ es igual a 82952 (Community, 2023).

 A triangle: 10 × 10 of type dbl

	0	1	2	3	4	5	6	7	8	9
1988	43393	82953	133351	179981	223988	253988	281159	308554	335588	362392
1989	23460	77938	128571	175509	209941	240347	270755	300549	328563	NA
1990	44843	85248	123765	153426	176985	200254	223067	245661	NA	NA
1991	40715	85394	120904	154776	187489	218360	247094	NA	NA	NA
1992	55743	100052	133936	163659	194613	222278	NA	NA	NA	NA
1993	49446	80206	108047	145226	180913	NA	NA	NA	NA	NA
1994	33465	65903	104361	142206	NA	NA	NA	NA	NA	NA
1995	32837	65286	99936	NA	NA	NA	NA	NA	NA	NA
1996	34711	66646	NA	NA	NA	NA	NA	NA	NA	NA
1997	34107	NA	NA	NA	NA	NA	NA	NA	NA	NA

A partir de esta notación se define la siguiente fórmula para el cálculo de los factores de desarrollo, se muestra para el caso puntual del primer factor de desarrollo f_0 .

$$f_0 = \frac{\sum_{i=1}^{10-0-1} C_{i,0+1}}{\sum_{i=1}^{10-0-1} C_{i,0}} = 1,9788 \quad (2.1)$$

A triangle: 10 x 10 of type dbl

	0	1	2	3	4	5	6	7	8	9
1988	43393	82953	133351	179981	223988	253988	281159	308554	335588	362392
1989	23460	77938	128571	175509	209941	240347	270755	300549	328563	NA
1990	44843	85248	123765	153426	176985	200254	223067	245661	NA	NA
1991	40715	85394	120904	154776	187489	218360	247094	NA	NA	NA
1992	55743	100052	133936	163659	194613	222278	NA	NA	NA	NA
1993	49446	80206	108047	145226	180913	NA	NA	NA	NA	NA
1994	33465	65903	104361	142206	NA	NA	NA	NA	NA	NA
1995	32837	65286	99936	NA	NA	NA	NA	NA	NA	NA
1996	34711	66646	NA	NA	NA	NA	NA	NA	NA	NA
1997	34107	NA	NA	NA	NA	NA	NA	NA	NA	NA

Es importante recalcar que el supuesto detrás de los factores de desarrollo es que se parte de la idea de que el crecimiento pasado determina el crecimiento futuro (Community, 2023).

Una vez se tienen los factores de desarrollo por periodo, se van multiplicando estos por el último valor conocido y así sucesivamente se va completando el triángulo inferior.

$$C_{1997,1} = C_{1997,0} * f_0 = 67491 \quad (2.2)$$

A matrix: 10 x 11 of type dbl

	0	1	2	3	4	5	6	7	8	9	ult
1988	43393	82953	133351	179981	223988	253988	281159	308554	335588	362392	429292
1989	23460	77938	128571	175509	209941	240347	270755	300549	328563	354806	420305
1990	44843	85248	123765	153426	176985	200254	223067	245661	267863	289257	342656
1991	40715	85394	120904	154776	187489	218360	247094	272532	297162	320897	380136
1992	55743	100052	133936	163659	194613	222278	248847	274466	299271	323174	382834
1993	49446	80206	108047	145226	180913	206822	231543	255380	278461	300702	356213
1994	33465	65903	104361	142206	171647	196229	219684	242300	264198	285300	337968
1995	32837	65286	99936	130616	157657	180236	201779	222552	242666	262048	310423
1996	34711	66646	98767	129088	155813	178127	199419	219948	239826	258982	306791
1997	34107	67491	100019	130725	157789	180386	201948	222738	242868	262266	310682

Finalmente si a la última columna si la última columna conocida como ultimate se le resta la diagonal (lo último conocido) se obtiene el IBNR.

Para el caso de Promutual estos fueron los resultados obtenidos, recordando que el valor real es el que se tenía desde el principio del ejercicio pero se partió del supuesto de que no lo conocíamos.

A data.frame: 1 x 3

Real	Estimado	Diferencia
<dbl>	<dbl>	<dbl>
1377449	1647504	-0.1960544

Figura 2.2: Resultados por Chain Ladder.

2.2. Regresión Lineal Chain Ladder.

De la metodología de Chain Ladder se observa que la dependencia viene directamente de la proporción de crecimiento de un periodo de desarrollo a otro, es decir de una relación entre columnas, sin embargo una forma alternativa de intentar predecir los valores de incurridos es considerando una dependencia entre filas y columnas .

De lo anterior surge un modelo multiplicativo que parte de la idea de que cada valor de incurridos del triángulo inferior se puede predecir como la multiplicación de dos variables llamadas U y S , donde U itera sobre las filas y S sobre las columnas (f, s.f.-b) .

$$U_i = \text{Parametro que recorer las filas.} \quad (2.3)$$

$$S_j = \text{Parametro que recorer las columnas.} \quad (2.4)$$

$$\begin{array}{c}
 S_j \\
 \begin{array}{c|ccc}
 & X_{11} & X_{12} & X_{13} \\
 U_i & X_{21} & X_{22} & \cancel{X_{23}} \\
 & X_{31} & \cancel{X_{32}} & \cancel{X_{33}}
 \end{array} \\
 \cancel{X}_{32} = U_3 \cdot S_2 \\
 \cancel{X}_{33} = U_3 \cdot S_3 \\
 \cancel{X}_{23} = U_2 \cdot S_3
 \end{array}$$

Del razonamiento anterior, se escribe la siguiente expresión:

$$X_{i,j} = U_i * S_j \quad (2.5)$$

Para simplificar la forma en la que se puede plantear esta multiplicación, se hace la siguiente simplificación:

$$\sum_{j=1}^t S_j = 1 \quad (2.6)$$

Bajo esta simplificación la interpretación de estas variables es más clara.

$$U_i = \text{Ultimate total esperado por año de accidente } i. \quad (2.7)$$

$$S_j = \text{Porcentaje esperado del ultimate que ocurre en el periodo de desarrollo } j. \quad (2.8)$$

Para poder convertir este problema multiplicativo en uno que se pueda resolver mediante regresión lineal se procede a sacar logaritmos a ambos lados de la ecuación 2.5 y se añade una media μ como intercepto de la regresión logarítmica. (f, s.f.-b)

$$\begin{aligned} \text{Log}(S_j) &= \beta_j \\ \text{Log}(U_i) &= \alpha_i \end{aligned}$$

$$\begin{aligned} E(Y_{ij}) &= \mu + \alpha_i + \beta_j \\ Y_{ij} &= \log(X_{ij}) \end{aligned}$$

De esta forma, es posible plantear tantas ecuaciones como siniestros conocidos se tienen, es decir tantos datos como tenga el triángulo superior, para simplificar la cantidad de parámetros se parte del supuesto de que $\alpha_1 = \beta_1 = 0$ para finalmente llegar siguientes ecuaciones:

$$\begin{aligned}
 Y_{11} &= \mu + \cancel{\alpha_1} + \cancel{\beta_1} \\
 Y_{12} &= \mu + \cancel{\alpha_1} + \beta_2 \\
 Y_{21} &= \mu + \alpha_2 + \cancel{\beta_1} \\
 Y_{13} &= \mu + \cancel{\alpha_1} + \beta_3 \\
 Y_{22} &= \mu + \alpha_2 + \beta_2 \\
 Y_{31} &= \mu + \alpha_3 + \cancel{\beta_1} \\
 \hline
 \alpha_1 + \beta_1 &= 0
 \end{aligned}$$

Las ecuaciones anteriores se pueden expresar en forma matricial.

$$\begin{bmatrix} Y_{11} \\ Y_{12} \\ Y_{21} \\ Y_{13} \\ Y_{22} \\ Y_{31} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 \\ 1 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} \mu \\ \alpha_2 \\ \alpha_3 \\ \beta_2 \\ \beta_3 \end{bmatrix}$$

Finalmente, se tiene una expresión lineal en la cual se como los y , los cuales son los logaritmos de los valores incurridos y los X que corresponden a la matriz de ceros y unos, por lo que el vector de parámetros α y β es posible estimar a partir de mínimos cuadrados, entre otros posibles métodos. (FS.S, 1989)

$$\hat{\beta} = (X^T * X)^{-1} * X^T Y \quad (2.9)$$

A las ecuaciones antes planteadas es importante calcular un término de error.

2.3. Lasso y Ridge.

Del modelo de regresión múltiple planteado anteriormente, surge la pregunta de si todos los coeficientes son relevantes para el modelo. La presencia de coeficientes no relevantes puede complicar la interpretación del modelo y dificultar su capacidad predictiva.

Por otro lado, se suma el hecho de que una fuerte correlación lineal entre variables explicativas provoca baja precisión en la estimación de los parámetros, pues resulta difícil aislar los efectos individuales de los β .

Es por ello que surgen modelos de regularización, como Ridge y Lasso, que buscan que los coeficientes tiendan a cero, minimizando así los riesgos de overfitting. Estos modelos de penalización actúan directamente sobre la magnitud de los coeficientes (Rodrigo, última actualización Noviembre 2020) .

2.3.1. Ridge.

De la estadística se sabe que el error cuadrático medio (MSE) está definido para el caso de una regresión lineal, de la siguiente forma:

$$MSE = \frac{1}{n} \|\vec{y} - \vec{x}\beta\|^2 \quad (2.10)$$

Al momento de querer encontrar los parámetros que mejor estimen los y se espera que el MSE sea lo menor posible.

Siendo el MSE un criterio de selección para determinar si un modelo es bueno o malo, este criterio por sí solo no toma en cuenta el impacto de posible correlación entre las variables, por lo que surge la necesidad de añadir un término que castigue el número de parámetros (variables) y haga tender a cero aquellas variables que pueden no estar aportando mucho al modelo (Kobak, 2021).

La siguiente ecuación da solución al problema antes planteado:

$$L = \frac{1}{n} \|\mathbf{y} - \mathbf{X}\beta\|^2 + \lambda R(\beta) \quad (2.11)$$

La expresión anterior se le conoce como función de pérdida.

El término en rojo de la expresión anterior (2.11), es el término de parametrización y puede ser contruido de las siguientes dos formas:

$$\lambda R(\beta) = \lambda \|\beta\|^2 = \lambda \sum_i \beta_i^2 \quad (2.12)$$

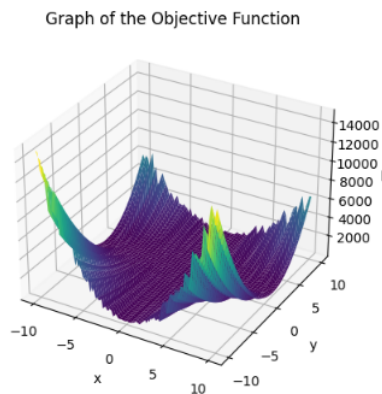
$$\lambda R(\beta) = \lambda \|\beta\|_1 = \lambda \sum_i |\beta_i| \quad (2.13)$$

La ecuación (2.12) corresponde a la regularización de Ridge y la (2.13) corresponde a la de lasso. El sub-índice en la regresión lasso hace referencia a que es la norma del valor absoluto (Kobak, 2021).

El λ es el parámetro de regularización, es una variable móvil que puede tomar varios valores y al final la idea es quedarse con el mejor estimador de λ , es decir el que disminuya la función de pérdidas. El método para calcular el λ es por medio de la validación cruzada

Una vez se tiene seleccionado el λ , el proceso para encontrar los mejores parámetros para los datos de "train" se obtiene de derivar con respecto a β la ecuación 2.11, que al ser una ecuación multivariada el proceso correcto consiste en obtener el gradiente.

Estas cuasiones multivariadas tiene la siguiente forma (Kobak, 2021):



$$\nabla L = -\frac{2}{n} \mathbf{X}^T (\mathbf{Y} - \mathbf{X}\beta) + 2\lambda\beta \quad (2.14)$$

El siguiente paso consiste en igualar a cero y despejar β , para así encontrar los valores de β que minimizar la expresión.

$$0 = -\frac{2}{n}\mathbf{X}^T(\mathbf{Y} - \mathbf{X}\hat{\beta}) + 2\lambda\hat{\beta} \quad (2.15)$$

$$\mathbf{X}^T\mathbf{Y} = \mathbf{X}^T\mathbf{X}\hat{\beta} + n\lambda\hat{\beta} \quad (2.16)$$

$$\mathbf{X}^T\mathbf{Y} = (\mathbf{X}^T\mathbf{X} + n\lambda\mathbf{I})\hat{\beta} \quad (2.17)$$

$$\hat{\beta} = (\mathbf{X}^T\mathbf{X} + n\lambda\mathbf{I})^{-1}\mathbf{X}^T\mathbf{Y} \quad (2.18)$$

Se observa que cuando $\lambda = 0$ se retoma la expresión (2.9) para los parámetros de una regresión por mínimos cuadrados.

Finalmente se llega a lo que se conoce como "Shrinkage estimator", estimador que reduce la varianza de una estimación ajustándola hacia un valor central o en este caso hacia cero.

2.3.2. Lasso.

Para el caso de lasso la expresión para la penalización cambia un poco.

$$L = \frac{1}{n}\|\mathbf{y} - \mathbf{X}\beta\|^2 + \lambda\|\beta\|_1 \quad (2.19)$$

Por otro lado, en la regresión por Ridge la penalización hace que los coeficientes tiendan a cero más no los hace cero lo que puede seguir generando un problema en la interpretación, al conservar un número grande de parámetros. El problema anterior se soluciona usando la regresión lasso (Kobak, 2021).

Algo que si puede ser negativo de la regresión lasso es que no hay solución analítica para la expresión (2.19). La presencia de la norma en la función de costo introduce una propiedad no diferenciable en el problema de optimización, lo que complica el cálculo de las derivadas respecto a los parámetros. Esto hace que no sea posible encontrar una solución analítica cerrada mediante la igualación de derivadas a cero.

Ridge:

- Si tienes predictores altamente correlacionados, Ridge reduce la influencia de todos

ellos simultáneamente y de una manera proporcional. Ninguno de los predictores tiende a ser excluido completamente; todos contribuyen, pero en proporciones más pequeñas.

- En Ridge es como si hubiera un suavizando de la influencia de todas las variables correlacionadas.

Lasso:

- En cambio, si se usa lasso y se tiene predictores altamente correlacionados, es más probable que lasso seleccione uno de ellos y le dé todo el peso, excluyendo a los demás. lasso tiene una tendencia a realizar selecciones más drásticas y puede eliminar por completo algunas variables en presencia de correlación.
- lasso se puede ver como si se eligiera una de las variables correlacionadas e ignora "las demás. Además, si los datos de entrada cambian ligeramente (pequeñas perturbaciones), las soluciones de lasso pueden variar significativamente, especialmente cuando las variables están altamente correlacionadas. Esto significa que lasso puede ser inestable en presencia de correlaciones fuertes entre predictores.

2.4. Redes Neuronales.

Las redes Neuronales es aquel algoritmo que le permite al computador aprender conceptos complicados a través de la construcción de conceptos simples. Este algoritmo consiste puntualmente en procesos jerárquicos organizados puntuales llamados neuronas, inspiradas en las conexiones de las redes neuronales del cerebro humano (Kobak, 2021).

Las aplicaciones que tienen las redes neuronales son infinitas y sin duda, cada vez van a estar implementadas en más campos por lo que es fundamental empezar a evaluar su uso en las compañías.

Algunas implementaciones son las siguientes:

- Reconocimiento facial.
- clasificación de imágenes y videos.

- Generación de textos.
- Creación de videojuegos.
- Diagnósticos médicos al permitir el análisis de imágenes.
- Pronóstico de series temporales para predicciones financieras.
- Detección de fraudes en transacciones.
- Conducción Autónoma.
- Análisis de riesgos.
- Predicción de tendencias del mercado.

Como tal la configuración de una red neuronal no es en donde recae la ciencia de la red neuronal, pues la configuración son simplemente a través de cadenas de redes que se conectan de todas las formas posibles, la ciencia está en los datos que le proporcionamos de entrenamiento, si la información que introducimos para entrenar al modelo es muy buena y de alta calidad, cuando llegue la información de prueba el sabrá hacer lo que se le pide.

Básicamente la red neuronal funciona como una caja negra a la que se le muestran varias veces por ejemplo las características de un limón, como color, forma, tamaño etc.. y después se le muestra una foto de cualquier fruta y la red debe ser capaz de distinguir cual es limón y cual no, si a esta misma red le preguntamos por manzanas claramente no sabrá responder pues su información de entrenamiento fueron limones.

A continuación, se muestra una gráfica muy disiente de la estructura de una red, para este caso puntual contiene dos cadenas cada una de cuatro neuronas y a la que le ingresan tres características o variables y sale en dos respuestas que para este caso son parámetros para la construcción de una regresión lineal o en el caso de un problema de clasificación sería los dos grupos de clasificación.

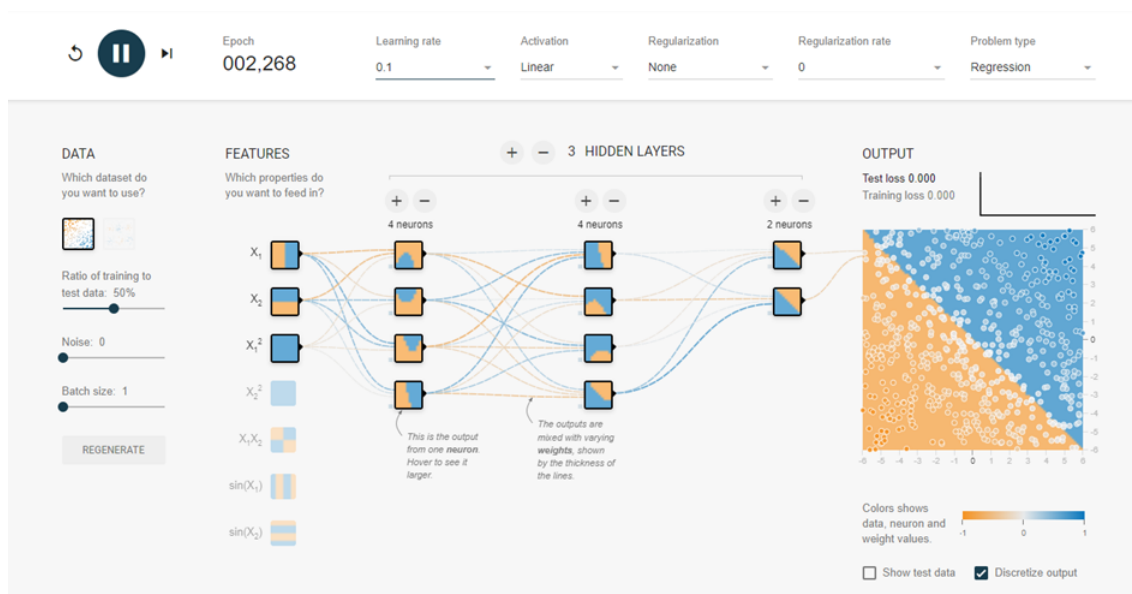


Figura 2.3: Gráfica realizada en playgroup

Capítulo 3

Datos.

La información utilizada para este proyecto proviene de un proyecto desarrollado por la CAS (Casualty Actuarial Society) en el que se recopiló un conjunto de datos correspondiente a triangulo de varias líneas de negocio personales y comerciales de las aseguradoras de propiedad y accidentes de los Estados Unidos más concretamente de del Programa de Análisis de Pérdidas y Gastos de la Asociación Nacional de Comisionados de Seguros (NAIC, por sus siglas en inglés).

Estos datos son de libre uso y proporcionan una base confiable para entrenar nuestro modelo más concretamente en el ramo de negligencia médica.

Esta información viene en una tabla grande con las siguientes columnas.(f, s.f.-a)

GRCODE Código de compañía NAIC (incluyendo grupos de aseguradoras y aseguradoras individuales)

GRNAME Nombre de la compañía NAIC (incluyendo grupos de aseguradoras y aseguradoras individuales)

AccidentYear Año del accidente (1988 a 1997)

DevelopmentYear Año de desarrollo (1988 a 1997)

DevelopmentLag Año de desarrollo (AY-1987 + DY-1987 - 1)

IncurLoss_ Pérdidas incurridas y gastos asignados reportados al final del año

CumPaidLoss_ Pérdidas acumuladas pagadas y gastos asignados al final del año

BulkLoss_ Reservas para pérdidas netas y gastos de defensa y control de costos informados al final del año

PostedReserve97_ Reservas publicadas en el año 1997 tomadas de la Exhibición de Suscripción e Inversiones - Parte 2A, incluyendo pérdidas no pagadas netas y gastos no pagados de ajuste de pérdidas

EarnedPremDIR_ Primas devengadas en el año de incurrimento - directas y asumidas

EarnedPremCeded_ Primas devengadas en el año de incurrimento - cedidas

EarnedPremNet_ Primas devengadas en el año de incurrimento - netas

Esta información de negligencia médica contiene la información de 34 aseguradoras, es decir que contiene 34 triángulo que incluyen reclamaciones de 10 años de accidentes (1988-1997) y 10 rezagos de desarrollo. Entre estos triángulos se encuentra el triángulo de Promutual (683), la empresa para la cual queremos desarrollar el modelo.

3.1. CRISP DM- Preparación de la data.

Esta información está originalmente en formato csv y en forma de tabla, por lo que lo primero que hay que hacer para entender la información es extraer las columnas de interés, las cuales son año de accidente, año de desarrollo e incurridos, esta es la información básica para construir los 34 triángulos.

La información de las tres columnas de interés es la siguiente

Año_Accidente	Año_desarrollo
Min. :1988	Min. :1988
1st Qu.:1990	1st Qu.:1994
Median :1992	Median :1997
Mean :1992	Mean :1997
3rd Qu.:1995	3rd Qu.:2000
Max. :1997	Max. :2006

```
Incurridos_año  
Min.   : -17  
1st Qu.:  0  
Median : 645  
Mean   : 11609  
3rd Qu.: 9050  
Max.   : 179425
```

También se hizo un cambio de nombre a las columnas para que fueran en español y más fácil de entender. Un paso muy importante en el proceso de preparación de data fue armar un vector que contenga todos los triángulos de arriba y otro con todos los triángulos de abajo, en otras palabras uno que contenga el desarrollo de los triángulos hasta 1997 y otro con la info de desarrollo hasta 2006.

Para el código que se encuentra anexado a este documento corresponde a los vectores:

resultados arriba
resultados abajo

Por otro lado se dejó apartada la aseguradora 683 correspondiente a Promutual.

Finalmente, la preparación de la data se resume en dos vectores cada uno con 34 triángulos, uno con los triángulos inferiores y otro con los superiores, muchos de estos triángulos contienen NA que más adelante serán reemplazados por cero.

3.2. CRISP DM- Entendimiento de la data.

El entendimiento de la data consistió en analizar los triángulos de las 33 aseguradoras restantes (dejando sólo la información con la que se va a entrenar mi modelo). Lo primero que se hizo fue crear 10 data frame con la información de incurridos y de años de desarrollo, es decir la información para las 33 aseguradoras de los incurridos ocurridos por ejemplo en 1988. Después se generaron 10 gráficas una por cada año en el que se muestra cómo fue el desarrollo de los siniestros en cada periodo y por cada aseguradora.

La siguiente imagen muestra la proporción de la reserva de IBNR para cada una de las 34 aseguradas. Para la creación de la gráfica se realizó un filtro por los años de desarrollo mayor a 1998 para poder extraer la información del triángulo inferior. En la imagen se observa

resaltado en rojo la fracción correspondiente a Promutual.

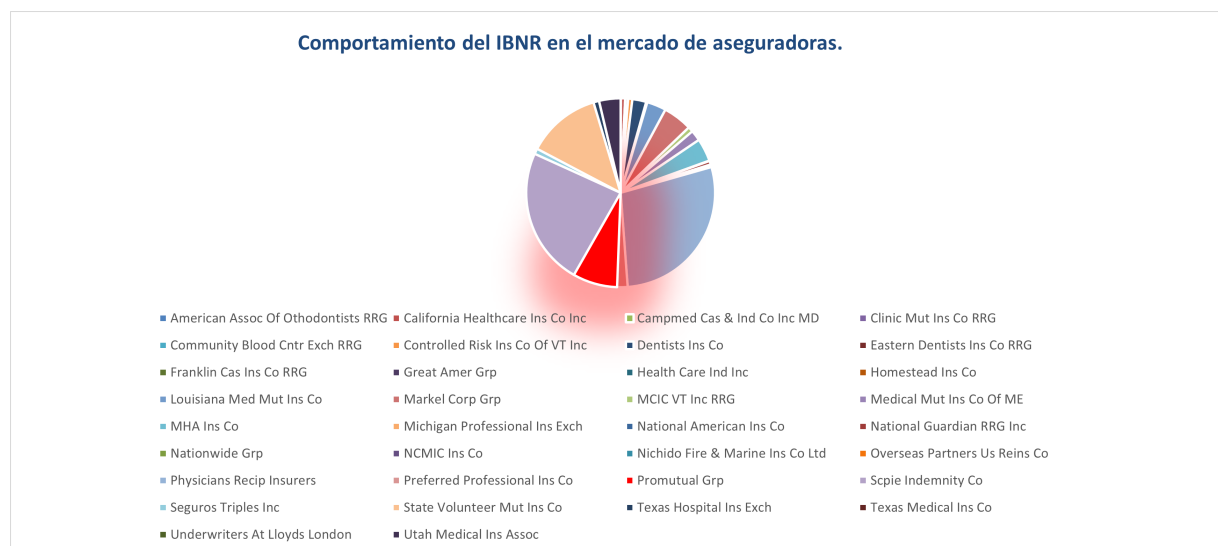


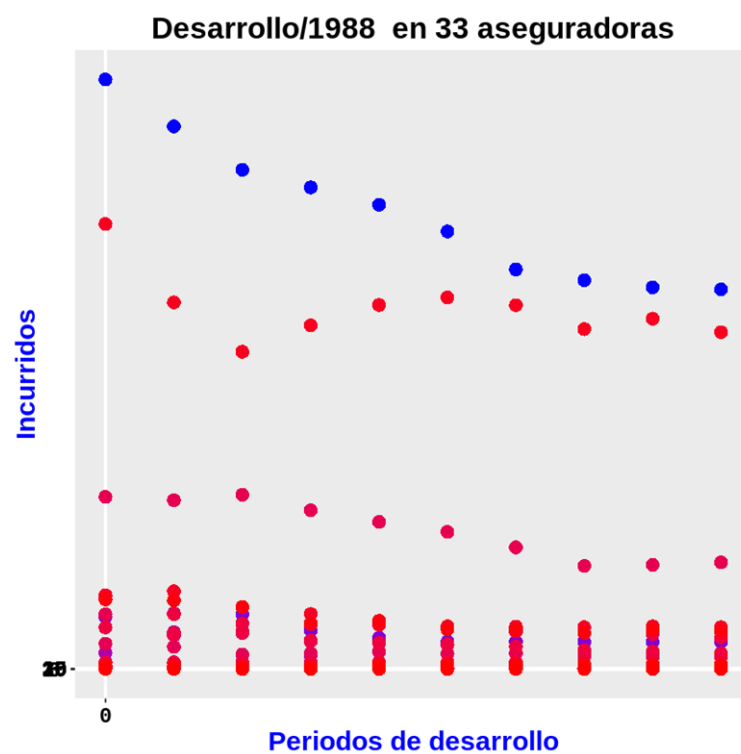
Figura 3.1: Comparativo de la reserva de IBNR en el mercado.

En esta otra representación visual (Figura 3.2), se exhibe de manera similar el valor de la reserva de IBNR para las 34 aseguradoras. Se destaca que Promutual presenta un valor notablemente superior al promedio. A pesar de esto, no figura entre las tres aseguradoras con las reservas más elevadas. La particularidad de esta gráfica radica en su capacidad para ilustrar el comportamiento singular de los seguros de negligencia médica, caracterizados por extremos. Esto se justifica por la posibilidad de que transcurran largos periodos, hasta diez años, sin que un hospital enfrente una demanda significativa en términos de monto. Sin embargo, la presencia de una sola demanda durante ese período puede aumentar drásticamente el promedio, generando picos inesperados en la gráfica.

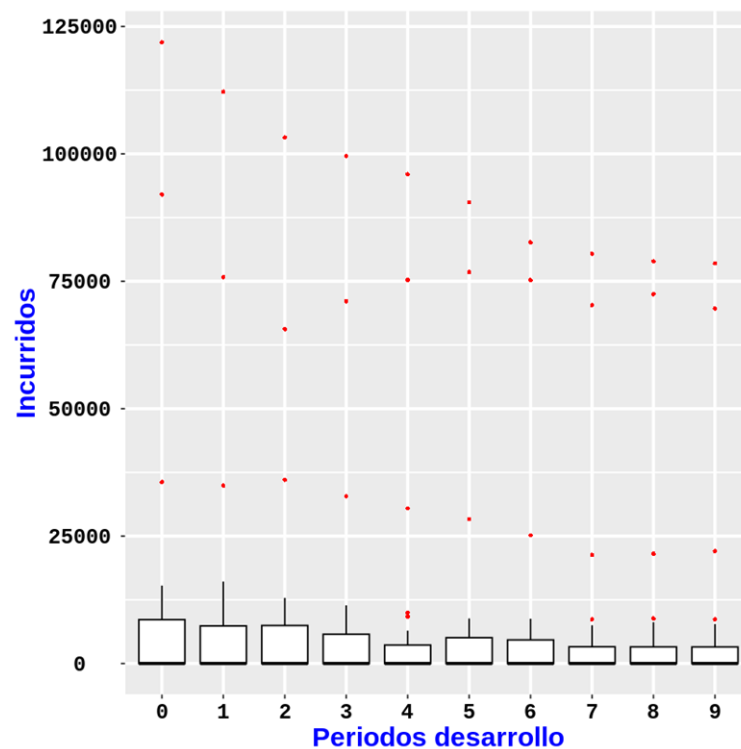
La siguiente gráfica se realizó para cada uno de los 10 años en donde se puede observar que hay aproximadamente 3 aseguradoras con un comportamiento diferente a las demás.



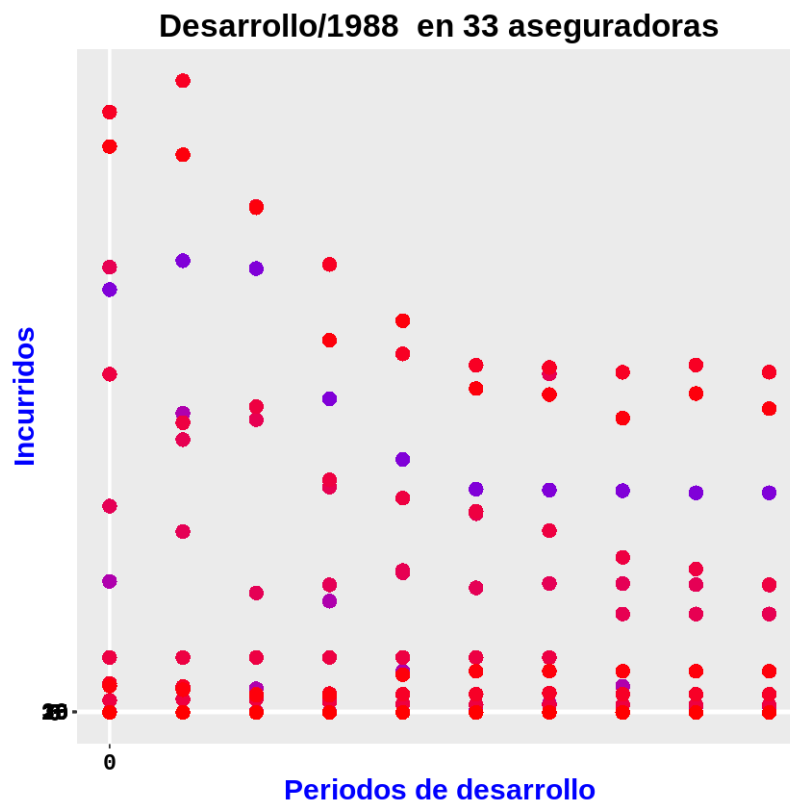
Figura 3.2: Comportamiento de las aseguradoras en el mercado.



Para comprobar cuáles son estas aseguradoras se hace un diagrama de cajas y se calculan los picos de los bigotes.



Las aseguradoras encontradas en las dies gráficas fueron las siguientes: 669,33049, 36676 y 41467, al eliminar estas aseguradoras de la data, la gráfica por ejemplo para 1988 queda de la siguiente forma.



Se observa que los datos se encuentran más juntos a lo largo del desarrollo de los siniestros.

El análisis anterior es de gran utilidad para entender el comportamiento de estas aseguradoras en el sector de negligencia médica y poder determinar cuáles de ellas no tienen un comportamiento similar al promedio, sin embargo esta no es una razón válida para excluir estos datos del análisis por machine learning.

Capítulo 4

Entrenamiento del modelo.

4.1. CRISP DM- Modelado.

Cuando se quiere poner a prueba diferentes modelos usando Machine learning es primordial la partición de los datos y la validación cruzada que se quiera implementar.

4.2. Partición de datos.

Este proyecto cuenta con los datos de 34 aseguradoras, sin embargo cada aseguradora cuenta con un triángulo superior y otro inferior por lo que en total hay 68 triángulos. Los triángulos con lo que se va a entrenar cada modelo son los reales o conocidos, es decir los 34 triángulos superiores, los triángulos inferiores servirán al momento de poner a prueba el modelo y obtener un criterio de calificación como MAPE o MSE.

De los 34 triángulos superiores hay 19 triángulos que no tienen pagos o accidentes en todos los años, como queremos un modelo que en particular funcione para la empresa de interés Promutual, se decidió quitar a aquellos triángulos que no asimilaban muy bien el comportamiento de promutual, un ejemplo de este tipo de triángulo es el siguiente:

17. A triangle: 10 × 10 of type dbl

	0	1	2	3	4	5	6	7	8	9
1988	0	0	0	0	0	0	0	0	0	0
1989	0	0	0	0	0	0	0	0	0	NA
1990	0	0	0	0	0	0	0	0	NA	NA
1991	0	0	0	0	0	0	0	NA	NA	NA
1992	0	0	0	0	0	0	NA	NA	NA	NA
1993	0	0	0	0	0	NA	NA	NA	NA	NA
1994	0	0	0	0	NA	NA	NA	NA	NA	NA
1995	0	0	0	NA	NA	NA	NA	NA	NA	NA
1996	52	221	NA	NA	NA	NA	NA	NA	NA	NA
1997	112	NA	NA	NA	NA	NA	NA	NA	NA	NA

De los 34 triángulos quedó un total de 15 triángulos, de estos 15 triángulos se aparta un triángulo al azar, este triángulo será el triángulo de test, esto se hace tanto para el conjunto de triángulos superiores como inferiores. La idea de apartar un dato del conjunto total es porque no se puede entrenar un modelo con el dato con el que se pondrá a prueba el modelo para impedir que el modelo esté condicionado y se aprenda el resultado “overfitting”. En la siguiente imagen se muestra el código con el que se apartó al azar un triángulo de la base de triángulos inferiores.

Aparto mi grupo de test

```

class(RESULTADOS_limpio)
indice_aleatorio <- sample(1:length_1, size = 1)
elemento_test_sup <- RESULTADOS_limpio[indice_aleatorio]
elemento_test_inf <- RESULTADOS_limpio_inf[indice_aleatorio]
elemento_test_inf

```

'list'

A matrix: 10 × 10 of type dbl

	0	1	2	3	4	5	6	7	8	9
1988	0	0	0	0	0	0	0	0	0	0
1989	0	0	0	0	0	0	0	0	0	4595
1990	0	0	0	0	0	0	0	0	6629	6629
1991	0	0	0	0	0	0	0	6305	6805	6805
1992	0	0	0	0	0	0	11740	11240	11240	10443
1993	0	0	0	0	0	13458	13458	13458	10915	10915
1994	0	0	0	0	19078	19076	19075	18186	18041	17988
1995	0	0	0	18533	18534	18534	16045	15639	15758	15732
1996	0	0	21579	21580	18579	13381	12494	12464	12308	12225
1997	0	22346	22346	25346	24659	23155	22606	22820	22525	22784

Con el grupo final de los 14 triángulos se generan varios grupos en los cuales poder

entrenar el modelo e ir evaluando resultados, es decir que es necesario crear otro grupo el de validación, la forma de hacer esto fue por combinatoria, crear todos los posibles grupos que se puedes armar con 14 datos apartando dos.

The image shows a digital calculator interface with a light green background. At the top, the word "Result" is displayed in blue. Below it, a horizontal line separates the header from the input section. The input section is labeled "Input:" in blue and shows "n = 14, r = 2". To the right of the input, there is a light purple box containing the word "Answer:" in blue and the number "91" in green. Below the input section, the "Solution:" section is shown in blue. It contains the formula $C(n,r) = n! / (r!(n-r)!)$ and a step-by-step calculation for n=14 and r=2, resulting in 91.

Result

Input:
n = 14, r = 2

Answer:
91

Solution:
Formula: $C(n,r) = n! / (r!(n-r)!)$

In this exapmle we have n = 14 and r = 2. by Putting these values in above formula, we have:

$$\begin{aligned} C(n,r) &= ? \\ C(n,r) &= C(14,2) \\ &= 14! / (2!(14 - 2)!) \\ &= 14! / 2! \times 12! \\ &= 91 \end{aligned}$$

4.3. Validación cruzada.

La validación cruzada es una técnica utilizada en estadísticas y aprendizaje automático para evaluar el rendimiento de un modelo predictivo y reducir el riesgo de sobreajuste (overfitting). El objetivo es dividir el conjunto de datos en subconjuntos de entrenamiento y prueba de manera repetida, lo que permite evaluar el rendimiento del modelo en diferentes conjuntos de datos y proporciona una estimación más robusta de su capacidad predictiva.

Con lo mostrado en la anterior imagen, se muestra que se están generando 91 combinaciones diferentes en las que podemos poner a prueba el modelo y en cada una de estas combinaciones se están apartando dos datos que sirven para el proceso de validación, el cual es un proceso de prueba previo al test.

Capítulo 5

Implementacion.

Como se mostró en el capítulo anterior la validación cruzada consiste en la construcción de un código que en cada iteración genere un grupo diferente de datos de entrenamiento y a estos datos de entrenamiento los ponga a funcionar en los respectivos modelos. Para este proyecto se construyó un for que genera la partición y que entrena los modelos de regresión lineal chain Ladder, regresión Ridge y regresión lasso, en cada iteración el código genera un criterio de selección , MAPE.

5.1. CRISP DM- Evaluado.

Según lo anterior, al final de la cadena de iteraciones se debe regenerar un data frame de 91 filas, donde se muestre el valor del MAPE para los tres podemos una última columna que selecciona el modelo que tuvo el menor MAPE. Es importante mencionar que el MAPE es este criterio de selección que califica qué tan bien le fue al modelo al ponerlo a prueba con el dato de evaluación, este dato de evaluación es el dato que se aparta en cada iteración, recordemos que en cada iteración se genera un grupo de 12 datos y se apartan 2.

Una vez se hizo el proceso de evaluación se vuelve a poner a prueba los modelos pero esta vez con los datos de test que en un inicio se apartaron y bajo le test se vuelve a obtener una tabla de resultados, esta tabla muestra directamente el mejor modelo y su MAPE.

5.1.1. Red Neuronal.

Para el proceso de la red neuronal se procedió de forma diferente, lo primero que se hizo fue definir la variable objetivo, los incurridos y como variables predictivas año de accidente y año de desarrollo. La red neuronal es secuencial con capas densas (totalmente conectadas).

A data.frame: 91 × 4

Regresion	Lasso	Ridge	MenorColumna
<I<list>>	<I<list>>	<I<list>>	<chr>
0.967497....	0.999938....	0.999941....	Regresion
0.965167....	0.999940....	0.999942....	Regresion
0.963870....	0.999940....	0.999942....	Regresion
0.964610....	0.999940....	0.999942....	Regresion
0.962203....	0.999942....	0.999944....	Regresion
0.962279....	0.999942....	0.999944....	Regresion
0.962487....	0.999941....	0.999944....	Regresion
0.961440....	0.999942....	0.999944....	Regresion
0.961795....	0.999942....	0.999944....	Regresion
0.962592....	0.999942....	0.999944....	Regresion

Figura 5.1: Ciclo Validación.

A data.frame: 91 × 3

TipoModelo	Resultados	Indicador
<chr>	<dbl>	<int>
Regresion	0.9999386	1
Regresion	0.9999405	2
Regresion	0.9999402	3
Regresion	0.9999405	4
Regresion	0.9999424	5
Regresion	0.9999421	6
Regresion	0.9999419	7
Regresion	0.9999425	8

Figura 5.2: Ciclo Test.

La red tiene una capa de entrada con 2 nodos, capas ocultas con activación ReLU y una capa de salida con activación lineal. Para la red neuronal el criterio de selección fue MSE y se configuró por dentro la validación cruzada dejando un dato por fuera. Más adelante se hará una explicación más detallada del modelo.

5.2. CRISP DM- Desarrollo.

5.2.1. Modelos Determinístico.

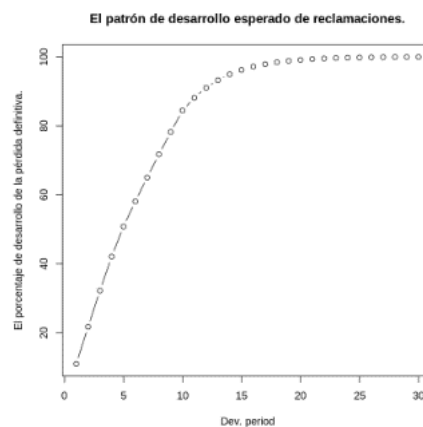
Para la programación del método determinístico, los pasos fueron los siguientes:

- Llamar al triángulo superior de Promutual y acumularlo.
- Generar los factores de desarrollo, este procedimiento se hizo en R pero también es IBNRS el software que ofrece la consultora de ADDACTIS.

```
[ ] n <- 10
factores_desarrollo <- sapply(1:(n-1),
  function(i){ # i avanza por las columnas
    sum(Promutual_acumulativo[c(1:(n-i)),i+1])/sum(Promutual_acumulativo[c(1:(n-i)),i])
  }
)
factores_desarrollo # estos factores representan el aumento entre años de desarrollo (para este caso 9)
```

1.97880723788597 · 1.48196055864879 · 1.30699642997415 · 1.20702936631238 · 1.14321118692952 · 1.11953132102669 · 1.10294833034616 · 1.09037551941133 · 1.07987174749991

- Después se extrapoló para calcular el último factor de desarrollo.
- La siguiente imagen muestra como fue el porcentaje de desarrollo a través del tiempo.



- Finalmente con los factores de desarrollo se calcula la parte de abajo de triángulo, es decir la reserva de IBNR.

```
factores_desarrollo <- c(factores_desarrollo , factores_desarrollo.tail)
fullPromutual_acumulativo <- cbind(Promutual_acumulativo, Ult = rep(0, 10))
for(k in 1:n){
  fullPromutual_acumulativo[(n-k+1):n, k+1] <- fullPromutual_acumulativo[(n-k+1):n,k]*factores_desarrollo [k]
}
round(fullPromutual_acumulativo)
```

A matrix: 10 × 11 of type dbl

	0	1	2	3	4	5	6	7	8	9	Ult
1988	43393	82953	133351	179981	223988	253988	281159	308554	335588	362392	429292
1989	23460	77938	128571	175509	209941	240347	270755	300549	328563	354806	420305
1990	44843	85248	123765	153426	176985	200254	223067	245661	267863	289257	342656
1991	40715	85394	120904	154776	187489	218360	247094	272532	297162	320897	380136
1992	55743	100052	133936	163659	194613	222278	248847	274466	299271	323174	382834
1993	49446	80206	108047	145226	180913	206822	231543	255380	278461	300702	356213
1994	33465	65903	104361	142206	171647	196229	219684	242300	264198	285300	337968
1995	32837	65286	99936	130616	157657	180236	201779	222552	242666	262048	310423
1996	34711	66646	98767	129088	155813	178127	199419	219948	239826	258982	306791
1997	34107	67491	100019	130725	157789	180386	201948	222738	242868	262266	310682

5.2.2. Modelos de regresión.

- Cargar bibliotecas.

Se crean vectores vector incurridos base y vector **incurridos base sub** para almacenar los resultados. Se define el número de elementos (**n elementos a tomar**) a seleccionar en cada combinación.

```
vector_incurridos_base <- c()
```

```
vector_incurridos_base_sub<- c()
```

```
# Datos iniciales son el vector resultados arriba
```

```
# Datos iniciales: Resultados_limpios
```

```
n_elementos_a_tomar <- 2
```

```
contador <- 0
```

- Generar combinaciones:

Se utilizan las funciones `combn` para generar todas las combinaciones posibles de 2 elementos a partir de dos conjuntos de datos (**RESULTADOS limpio** y **Resultados limpio inf**).

```
combinaciones <- combn(RESULTADOS_limpio, n_elementos_a_tomar)
combinaciones_1 <- combn(Resultados_limpio_inf, n_elementos_a_tomar)
```

- Configurar listas y matrices para resultados:

Se crean listas y matrices para almacenar los resultados de regresión, Lasso y Ridge.

```
validacion_regresio <- vector("list", length = ncol(combinaciones))
parametros <- vector("list", length = ncol(combinaciones))
comparativo <- vector("list", length = ncol(combinaciones))

comparativo_lasso <- vector("list", length = ncol(combinaciones))
parametros_LASSO <- vector("list", length = ncol(combinaciones))
validacion_LASSO <- vector("list", length = ncol(combinaciones))

parametros_RIDGE <- vector("list", length = ncol(combinaciones))
validacion_RIDGE <- vector("list", length = ncol(combinaciones))
comparativo_RIDGE <- vector("list", length = ncol(combinaciones))
```

- Bucle principal para procesar combinaciones:

- Se recorren las combinaciones y se realizan los siguientes pasos para cada combinación:
- Se obtienen los elementos de prueba y entrenamiento.
- Se generan vectores **vector incurridos base** y **vector incurridos base sub** mediante bucles anidados.
- Se reemplazan los valores infinitos por 0 en **Incurrido limpio2**.

- Creación de la matriz MATRIZ X:

Se crea una matriz apilada (**matriz apilada**) y se concatena para formar la matriz **MATRIZ X**. Se eliminan las filas con valores faltantes.

- Transformaciones y cálculo de parámetros:
 - Se realiza una serie de transformaciones y cálculos de parámetros utilizando regresión clásica.
 - Los parámetros se almacenan en la lista `parametros`.
- Prueba del modelo y evaluación de criterios:

```

#### saco el IBNR calculado por regresion y usado con dato de validacion ####
validacion_regresio[[i]] <- sum(matriz4 %% parametros[[i]])

# criterio de evaluación #####
### HAGO CRITERIO RESPECTO A MI RESERVA TOTAL (n=1) ###
comparativo[[i]] <- data.frame(
  Estimado = as.numeric(validacion_regresio[[i]]),
  Real = IBNR_guardado_vector_2,
  MAPE = abs(IBNR_guardado_vector_2 - as.numeric(validacion_regresio[[i]]))/IBNR_guardado_vector_2
)

mape_lista <- lapply(comparativo, function(df) df$MAPE)

```

- Lasso y Ridge:
 - Se utiliza el paquete `glmnet` para realizar regresiones Lasso y Ridge.
 - Se selecciona el mejor λ a través de la validación cruzada.
 - Se obtienen los mejores coeficientes y se calculan los IBNR estimados para Lasso y Ridge.
 - Se evalúan los resultados utilizando el MAPE.
- Construcción de un marco de datos de comparación:
 - Se construye un marco de datos (Comparativo validacion) para comparar los resultados de regresión, Lasso y Ridge.
 - Se identifica el método con el menor MAPE para cada combinación.
- Resultado final:
 - Se imprime y devuelve el marco de datos Comparativo validacion, que proporciona una comparación de los métodos de regresión, Lasso y Ridge en términos de MAPE para diferentes combinaciones de variables.

Este código es bastante extenso y realiza un análisis detallado de diferentes enfoques para la estimación de IBNR en el contexto de seguros.

5.2.3. Red Neuronal.

Este código en R utiliza la biblioteca keras para construir, compilar y entrenar un modelo de red neuronal para regresión. A continuación, te proporciono una explicación detallada del código:

- Cargar bibliotecas.

```
library(readr)
library(dplyr)
library(keras)
library(caret)
```

Se cargan las bibliotecas necesarias: readr para leer datos, dplyr para manipulación de datos, keras para construir modelos de red neuronal y caret para configurar la validación cruzada.

- Leer el archivo CSV y eliminar filas con valores faltantes:

```
datos <- read.csv("medmal_pos.csv")
datos <- na.omit(datos)
```

Lee un archivo CSV llamado medmal_pos.csv y elimina las filas que contienen valores faltantes.

- Crear variables objetivo y atributos predictivos:

```
y <- datos$IncurLoss_F2
X <- datos[, c("AccidentYear", "DevelopmentYear")]
```

efine la variable objetivo (y), que es **IncurLoss F2**, y los atributos predictivos (X), que son **AccidentYear** y **DevelopmentYear**.

- Normalizar los datos:

```
X <- scale(X)
y <- scale(y)
```

Normaliza los atributos predictivos y la variable objetivo para asegurarse de que tengan una escala similar.

- Crear el modelo de red neuronal con Keras:

```
model <- keras_model_sequential() %>%  
  layer_dense(units = 64, activation = "relu", input_shape = c(2)) %>%  
  # Capas ocultas con activación relu  
  layer_dense(units = 32, activation = "relu") %>%  
  layer_dense(units = 16, activation = "relu") %>%  
  layer_dense(units = 8, activation = "relu") %>%  
  layer_dense(units = 4, activation = "relu") %>%  
  # Capa de salida con activación lineal  
  layer_dense(units = 1, activation = "linear")
```

Crea un modelo de red neuronal secuencial con capas densas (totalmente conectadas). La red tiene una capa de entrada con 2 nodos, capas ocultas con activación ReLU y una capa de salida con activación lineal.

- Compilar el modelo:

```
model %>% compile(  
  optimizer = optimizer_adam(),  
  loss = "mean_squared_error",  
  metrics = c("mean_squared_error")  
)
```

Compila el modelo con el optimizador Adam, la función de pérdida de error cuadrático medio (mean squared error) y la métrica de error cuadrático medio.

- Dividir los datos en conjuntos de entrenamiento y prueba:

```
set.seed(123)  
index <- sample(1:nrow(datos), nrow(datos)*0.8)
```

```
train_data <- datos[index, ]
test_data <- datos[-index, ]
X_train <- scale(train_data[, c("AccidentYear", "DevelopmentYear")])
y_train <- scale(train_data$IncurLoss_F2)
X_test <- scale(test_data[, c("AccidentYear", "DevelopmentYear")])
y_test <- scale(test_data$IncurLoss_F2)
```

Establece una semilla para reproducibilidad y divide los datos en conjuntos de entrenamiento (**train data**) y prueba (**test data**). Normaliza los datos de entrenamiento y prueba.

- Especificar el método de validación cruzada LOOCV:

```
ctrl <- trainControl(method = "LOOCV")
```

Configura el método de validación cruzada dejando uno fuera (LOOCV) utilizando caret.

- Configurar callbacks para el modelo:

```
callbacks <- list(
  callback_early_stopping(patience = 10),
  callback_model_checkpoint("model_checkpoint.h5", save_best_only = TRUE),
  callback_tensorboard("logs", update_freq = "epoch")
)
```


Configura callbacks, como detener el entrenamiento temprano, guardar el mejor modelo y utilizar TensorBoard para el registro.

- Ajustar el modelo con los datos de entrenamiento:

```
fit <- model %>% fit(  
  x = as.matrix(X_train),  
  y = y_train,  
  epochs = 50,  
  batch_size = 20,  
  verbose = 1,  
  callbacks = callbacks,  
  validation_data = list(as.matrix(X_test), y_test),  
  trControl = ctrl  
)
```

Ajusta el modelo utilizando los datos de entrenamiento y especifica configuraciones como el número de épocas, tamaño del lote y callbacks. También utiliza validación cruzada LOOCV.

- Calcular el error cuadrático medio (MSE) para el modelo:

```
nn_avg_mse <- mean((predict(model, as.matrix(X_test)) - y_test)^2, na.rm = TRUE)
```

Calcula el MSE para el modelo utilizando los datos de prueba.

- Imprimir el resultado:

```
cat("Neural Network LOOCV-MSE:", nn_avg_mse, "\n")
```

Imprime el resultado del MSE obtenido mediante la validación cruzada LOOCV.

Capítulo 6

Resultados.

6.1. Regresión lineal y Chain Ladder.

El modelo determinístico de chain ladder y el modelo lineal de regresión se implementaron puntualmente para la aseguradora Promutual, los resultados obtenidos fueron los siguientes:

A data.frame: 1 × 3

Regresion	chain_ladder	Diferencia
<dbl>	<dbl>	<dbl>
2063966	1647504	-0.2527838

6.2. Validación.

Para el ciclo de validación se generó un data frame mostrado en la imagen (5.1) en este data frame todas las iteraciones marcaron que el mejor modelo es regresión y solo uno marcó que era lasso, pero entre todos el mejor MAPE lo tuvo el modelo de regresión.

A data.frame: 1 × 4

Regresion	Lasso	Ridge	MenorColumna
<I<list>>	<I<list>>	<I<list>>	<chr>
37 0.076494....	0.998653....	0.998698....	Regresion

6.3. Test.

Para el ciclo de test se generó un data frame mostrado en la imagen (5.2) en este data frame todas las iteraciones marcaron que el mejor modelo es regresión, pero en menor MAPE fue el siguiente:

```
A data.frame: 1 × 3
```

	TipoModelo	Resultados	Indicador
	<chr>	<dbl>	<int>
77	Regresion	0.9986416	77

6.4. Red Neuronal.

El MSE obtenido por red neuronal fue el siguiente:

```
Neural Network LOOCV-MSE: 0.9919287
```

Al comparar la reserba de IBNR calculadoa por redes neurales y la reserva real los resultados fueron los siguientes:

```
A data.frame: 1 × 3
```

Red_Neuronal	Real	Diferencia
<dbl>	<dbl>	<dbl>
1160934	1377449	-0.5913488

Figura 6.1: Resultados red neuronal.

6.5. Comparativos gnerales.

La siguiente imagen muestra los resultados del IBNR respecto a las diferentes metodologías como chain Ladder, regresión lineal, lasso (que presentó mejor desempeño que ridge) y redes neuronales, estas metodologías se comparan con el IBNR real de Promutual (triángulo inferior) y con el IBNR promedio del resto de las aseguradoras.

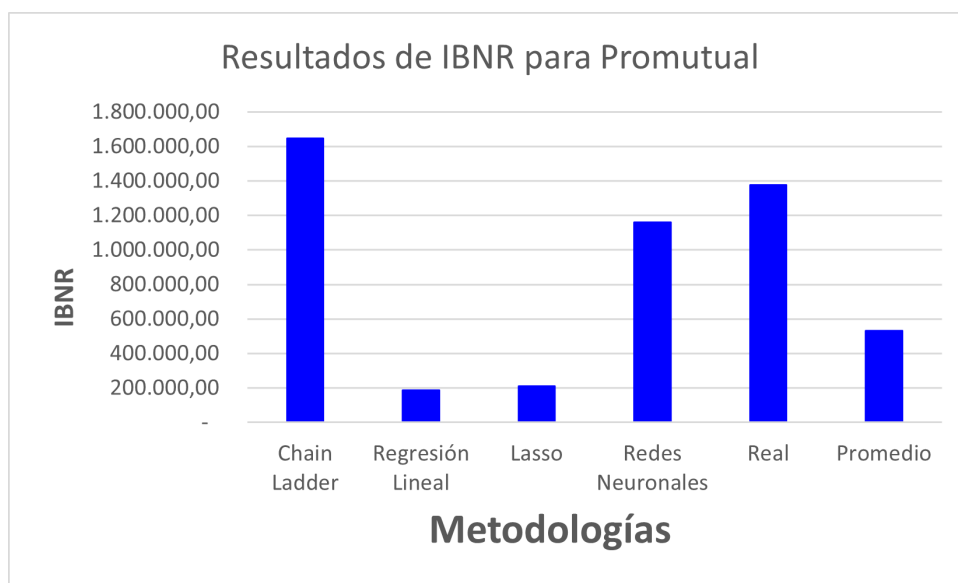


Figura 6.2: Comparativo de metodologías.

La siguiente imagen (6.3) muestra la diferencia porcentual de las metodologías con respecto al valor real que presentó Promutual, es decir que tal lejos estuvieron de replicar el resultado real.

Finalmente se observa que el mejor modelo aplicado a la data puntual de Promutual después del determinístico Chain Ladder (figura 2.1) es el de redes neuronales (figura 6.1) y le sigue el de regresión, sin embargo es importante mencionar que estos modelos sirven para el caso de Promutual pero es posible que para data extraña o de otra aseguradora el mejor modelo sea otro.

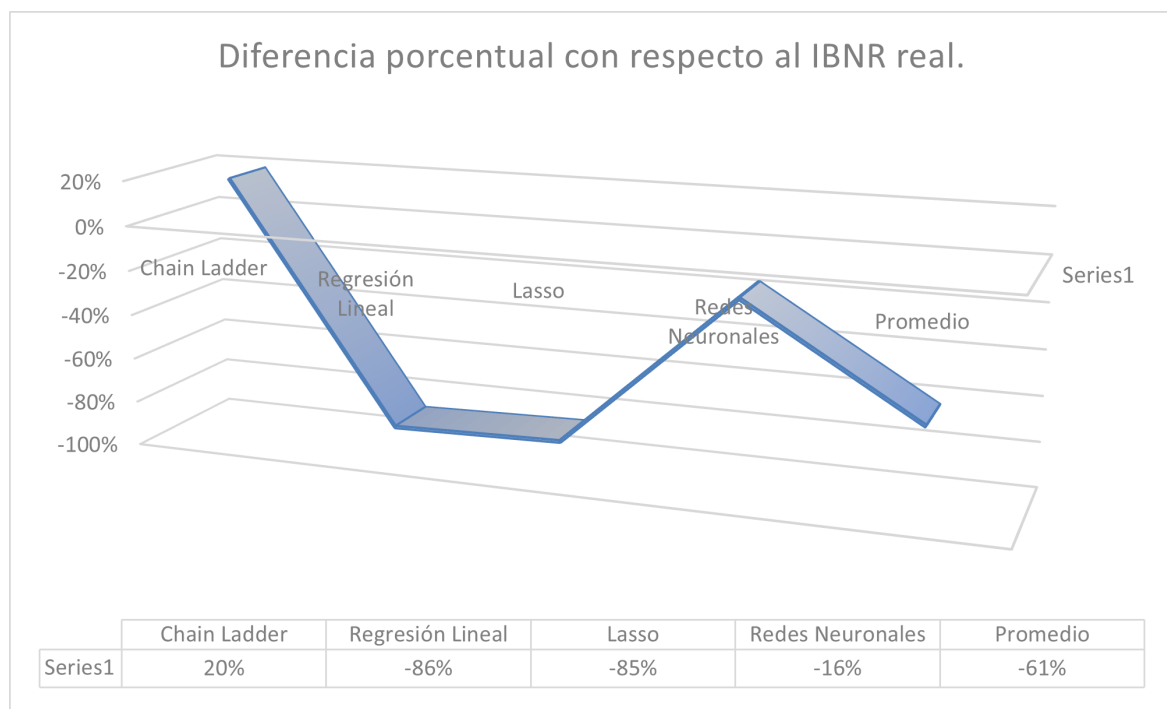


Figura 6.3: Diferencia porcentual con respecto al real.

Capítulo 7

Riesgos y Limitaciones.

Si bien el modelo determinístico demuestra eficacia al proyectar el futuro basándose en el historial más reciente de una compañía, su utilidad puede ser limitada en el contexto de aseguradoras de negligencia médica. Estas empresas, con frecuencia, carecen de un historial extenso, además de que el comportamiento resulta notablemente impredecible. En este sector, es común experimentar períodos prolongados de tranquilidad, seguidos un pico repentino provocado por alguna demanda que incluso pudo haber ocurrido hace años pero apenas resulta ser vigente ahora dados los resultados de un juez.

Por las razones expuestas, resulta sumamente beneficioso recurrir a la información del mercado, es decir, a datos provenientes de otras aseguradoras, con el fin de desarrollar modelos más robustos. En este enfoque, ya no se busca simplemente replicar el comportamiento histórico de Promutual, sino orientar el modelo hacia el comportamiento general del mercado. No obstante, es crucial poseer un profundo conocimiento del mercado en este contexto. Se corre el riesgo de utilizar datos de aseguradoras que difieren significativamente de nuestra aseguradora de interés. Por ejemplo, supongamos que la aseguradora B trabaja con reaseguradoras, es decir, que externaliza la cobertura de sus siniestros. Aunque la aseguradora B pueda asumir riesgos más elevados, como terremotos, debido a su práctica de reaseguro, emplear sus datos para entrenar un modelo destinado a prever el comportamiento de la aseguradora A podría generar problemas en la precisión de las predicciones.

Capítulo 8

Conclusiones

- Los resultados obtenidos por lasso, ridge y regresión lineal, son resultados más acordes al comportamiento de la mayoría de las aseguradoras del mercado de negligencia médica. En la imagen 3.2 se observa que la reserva de IBNR para más de la mitad de las aseguradoras está por debajo del millón. Esto significa que estos modelos que fueron entrenados con datos del mercado definitivamente tienen un comportamiento más acorde a este.
- Los resultados obtenidos por la red neuronal y el modelo determinístico replican mucho mejor el comportamiento real de Promutual. El modelo de redes neuronales utiliza toda la data del mercado pero consigue aproximarse más al real, esto se puede deber a que usa más de una variable explicativa y a la construcción particular.
- Por otro lado una conclusión importante es que el modelo de redes neuronales, a pesar de ser quizás el más complejo conceptualmente presentó muy buenos resultados, sin embargo se recomienda ponerlo a prueba constantemente e ir mejorando la red con más data.

Referencias

- Community, A. (2023). *Loss data analytics*. Loss Data Analytics. Descargado de <https://ewfrees.github.io/Loss-Data-Analytics-Spanish/>
- f, R. J. (s.f.-a). *Loss reserving data pulled from naic schedule p*.
- f, R. J. (s.f.-b). *Statistical methods for the chain ladder technique*.
- Facecolda. (2023). *Estimación de la reserva de siniestros avisados y no avisados*. Autor. Descargado de https://www.fasecolda.com/cms/wp-content/uploads/2019/08/15_efectos_en_el_pg_en_subestimar_la_reserva_de_ibnr.pdf
- E.S.S, R. . V. M. M. (1989). A state space representation of the chain ladder linear model. *City University, London*, 891–921.
- infobae. (2023). *Maya kowalski finalmente obtuvo justicia: un jurado de florida falló a su favor*. Avon. Descargado de <https://www.infobae.com/estados-unidos/2023/11/12/maya-kowalski-finalmente-obtuvo-justicia-un-jurado-de-florida-fallo-a-su-favor/>,
- Kobak, D. (2021). *Tübingen machine learning / introduction to machine learning*. Descargado de <https://dkobak.github.io/>
- Rodrigo, J. A. (última actualización Noviembre 2020). *Selección de predictores, regularización ridge, lasso, elastic net y reducción de dimensionalidad*. Descargado de https://cienciadedatos.net/documentos/31_seleccion_de_predictores_subset_selection_ridge_lasso_dimension_reduction