Máster en Ciencias Actuariales y Financieras

TRABAJO FINAL DE MÁSTER

Título: Cálculo de reservas con técnicas de *Machine Learning*

Autoría: Patricia López Ozcoz

Tutoría: Eva Boj y Mª Teresa Costa

Curso académico: 2024 - 2025

Universitat de Barcelona
Facultat d'Economia i Empresa

Màster de Ciències Actuarials i Financeres

Facultad de Economía y Empresa
Universidad de Barcelona

Trabajo Final de Máster
Máster en Ciencias Actuariales y Financieras

CÁLCULO DE RESERVAS CON TÉCNICAS DE MACHINE LEARNING

Autoría: Patricia López Ozcoz

Tutoría: Eva Boj y Mª Teresa Costa

El contenido de este documento es de exclusiva responsabilidad del autor, quien declara que no ha incurrido en plagio y que la totalidad de referencias a otros autores han sido expresadas en el texto.

RESUMEN

Este trabajo tiene como objetivo analizar y comparar distintas metodologías, centrándonos en las técnicas de *Machine Learning,* para estimar reservas de siniestros. Para ello, se implementan y evalúan varios modelos de *Machine Learning*, utilizando una base de datos representativa del sector asegurador. A lo largo del estudio, se realiza un exhaustivo preprocesamiento y validación para garantizar la robustez de los resultados. Las predicciones obtenidas se evalúan mediante métricas específicas, permitiendo identificar el modelo con mejor capacidad predictiva. Paralelamente, se calculan las reservas utilizando métodos tradicionales. Finalmente, se lleva a cabo una comparación entre ambos enfoques.

PALABRAS CLAVE

Reservas técnicas, Siniestros, *Machine Learning*, RBNS, IBNR, Chain-Ladder.

RESUM

Aquest treball té com a objectiu analitzar i comparar diferents metodologies, centrant-se en les tècniques de *Machine Learning*, per estimar les reserves de sinistres. Per això, s'implementen i s'avaluen diversos models de Machine Learning, utilitzant una base de dades representativa del sector assegurador. Al llarg de l'estudi, es realitza un exhaustiu preprocessament i una validació per garantir la robustesa dels resultats. Les prediccions obtingudes s'avaluen mitjançant mètriques específiques, fet que permet identificar el model amb millor capacitat predictiva. Paral·lelament, es calculen les reserves utilitzant mètodes tradicionals. Finalment, es fa una comparació entre ambdós enfocaments.

PARAULES CLAU

Reserves tècniques, Sinistres, *Machine Learning*, RBNS, IBNR, Chain-Ladder.

ÍNDICE

1. INTRODUCCIÓN

El cálculo de reservas es uno de los pilares del sector asegurador, ya que permite garantizar la estabilidad financiera de las compañías frente a los siniestros ocurridos y los pagos futuros derivados de éstos. En particular, el cálculo de reservas para siniestros de responsabilidad civil de vehículos es un área clave debido a la complejidad de los factores involucrados y la naturaleza impredecible de éstos en el ámbito del automóvil. Este tipo de siniestros, que involucra tanto la frecuencia de accidentes como la severidad de los daños ocasionados, requiere un análisis y preciso para asegurar que las compañías aseguradoras puedan cubrir los costes derivados de estos eventos

La estimación adecuada de las reservas de siniestros implica no solo una valoración precisa de los pagos futuros, sino también la capacidad de gestionar de forma eficiente un volumen considerable de datos históricos sobre accidentes y sus consecuencias. Los métodos actuariales tradicionales han sido la base sobre la que se han calculado estas reservas durante décadas. Métodos como Chain-Ladder (CL) han sido herramientas fundamentales en este proceso, permitiendo calcular reservas a partir de patrones históricos de siniestros y pagos. Sin embargo, estos enfoques presentan limitaciones notables cuando se enfrentan a datos complejos, no lineales y de gran volumen, características que son cada vez más comunes en la industria aseguradora actual.

Uno de los principales retos a los que se enfrentan los métodos actuariales clásicos es la incapacidad para capturar la no linealidad y las interacciones complejas entre las variables que influyen en el comportamiento de los siniestros. Además, estos métodos dependen de una serie de supuestos, como la distribución de los siniestros y los pagos futuros, que no siempre reflejan las realidades del entorno cambiante en el que operan las aseguradoras. En este contexto, la introducción de técnicas de *Machine Learning* (ML) ha supuesto una revolución en diversos campos, incluyendo el sector de los seguros, ya que ofrecen una mayor capacidad para manejar grandes volúmenes de datos y descubrir patrones complejos que pueden pasar desapercibidos para los métodos tradicionales.

Las técnicas de ML han demostrado ser herramientas poderosas para la predicción y el análisis de datos complejos. Estas técnicas permiten modelar relaciones no lineales entre las variables, adaptarse a nuevas tendencias y aprender directamente de los datos, sin la necesidad de hacer supuestos rígidos sobre la distribución de los siniestros o los pagos. Además, los modelos de ML tienen la capacidad de identificar patrones ocultos en los datos, lo que puede mejorar significativamente la precisión de las predicciones de las reservas y, por ende, la solvencia de las compañías aseguradoras.

Este trabajo tiene como objetivo evaluar la aplicabilidad de las técnicas de ML en el cálculo de reservas para siniestros de responsabilidad civil de vehículos por lesiones corporales, comparando estos métodos con los métodos actuariales tradicionales. En particular, se analiza si el uso de ML puede mejorar la precisión y eficiencia en la estimación de las reservas, ya que los modelos tradicionales no siempre consiguen capturar la complejidad inherente a los datos de siniestros.

El uso de ML en este ámbito no solo tiene implicaciones para mejorar la precisión del cálculo de reservas, sino que también puede tener un impacto en la forma en que las aseguradoras gestionan el riesgo en sus carteras. La capacidad para predecir con mayor exactitud los pagos futuros puede permitir a las aseguradoras optimizar la asignación de recursos, ajustar sus primas y, en última instancia, mejorar la rentabilidad. Además, el

uso de modelos interpretables como una segunda fase de modelización puede proporcionar a los profesionales del sector una visión más clara de los factores que influyen en los siniestros, lo que facilita la toma de decisiones informadas sobre la gestión de riesgos.

La importancia de este estudio radica en la necesidad de adaptar los métodos actuariales a los cambios tecnológicos y a la evolución del mercado asegurador. Con el aumento de la digitalización, la disponibilidad de grandes volúmenes de datos y el cambio en las dinámicas de los siniestros, las compañías de seguros necesitan herramientas más sofisticadas y precisas para gestionar el riesgo y garantizar su estabilidad financiera. En este sentido, el uso de ML puede representar una oportunidad para transformar la forma en que se calculan las reservas y, por ende, la manera en que las aseguradoras gestionan los riesgos asociados a los siniestros de responsabilidad civil.

Algunos trabajos recientes han explorado con éxito la aplicación de técnicas de ML al cálculo de reservas, comparándolas con métodos tradicionales como los modelos lineales generalizados (en inglés *generalized linear models,* GLM). Por ejemplo, Ahlgren (2018) analiza el uso del algoritmo de *Gradient Boosting* frente a GLM para siniestros reportados pero no liquidados (en inglés *reported but not settled,* RBNS), encontrando que, si bien el GLM presenta menor error cuadrático medio, el enfoque de *Gradient Boosting* ofrece una mayor flexibilidad para capturar dependencias complejas en los datos. Del mismo modo, Wüthrich (2018) investiga el uso de redes neuronales aplicadas al método CL, mientras que Duval y Pigeon (2019) proponen un modelo de reservas individuales basado en *Gradient Boosting*, resaltando la capacidad de este enfoque para mejorar la precisión en escenarios de datos complejos y heterogéneos.

Este trabajo se estructura en seis secciones principales. En la sección 1 se presenta el problema, los objetivos del estudio y la justificación del enfoque adoptado. La sección 2 incluye el marco teórico, donde se revisan los métodos tradicionales para el cálculo de reservas, incluyendo los triángulos de desarrollo y los modelos actuariales clásicos como CL y GLM, además de sus limitaciones. También en esta sección se aborda el uso de técnicas de ML para el cálculo de reservas, describiendo sus fundamentos matemáticos, modelos aplicables como árboles de decisión, *K-Nearest Neighbors* (KNN), *Lasso* y GLM penalizados, así como sus ventajas y desafíos. En la sección 3 se expone el marco práctico del estudio, detallando la generación de datos mediante el simulador *SynthETIC (*Avanzi et al., 2021), el análisis estadístico de la base de datos obtenida y la implementación de las distintas técnicas de ML, con la evaluación de su poder predictivo y la proyección de pagos futuros. Además, se incluyen los métodos tradicionales de cálculo de reservas aplicados para comparar resultados, y se presenta un análisis comparativo entre las técnicas estudiadas, destacando las diferencias y similitudes en las estimaciones obtenidas. En la sección 4 se recogen las conclusiones generales del trabajo. En las secciones 5 y 6 se incluyen la bibliografía consultada y los anexos, que contienen tablas, gráficos y el código R (R Development Core Team, 2025) utilizado para la implementación de los métodos.

2. MARCO TEÓRICO

El cálculo de reservas técnicas en seguros es una tarea fundamental desde el punto de vista financiero, actuarial y regulatorio. Las reservas representan los pasivos que una entidad aseguradora debe mantener para hacer frente a los compromisos derivados de siniestros ocurridos y no completamente liquidados.

En el caso concreto del seguro de responsabilidad civil de automóviles por lesiones corporales, el proceso de resolución de un siniestro puede prolongarse durante años debido a la naturaleza médica, jurídica y administrativa de este tipo de daños. Así, es frecuente que un siniestro se notifique con retraso, evolucione de forma no lineal y genere múltiples pagos parciales hasta su cierre definitivo. Esto plantea importantes retos para la estimación de reservas adecuadas.

Dentro de las reservas, es común distinguir tres componentes:

- RBNS: Reserva de siniestros ocurridos y notificados, pero no totalmente pagados (*Reported But Not Settled*).
- IBNR: Reserva de siniestros ocurridos, pero no notificados (*Incurred But Not Reported*).
- Reserva de gastos de gestión de siniestros (*claims handling expenses*).

El cálculo de estas reservas presenta una gran complejidad técnica debido a la incertidumbre inherente a la frecuencia, gravedad, tiempo de notificación y desarrollo de los siniestros. Esta incertidumbre ha dado lugar a la construcción de una extensa literatura técnica y metodológica que trata de dar respuesta a este problema desde distintos enfoques.

Para el cálculo de la reserva total en un seguro, se integran las tres componentes principales, ajustados a la realidad operativa y regulatoria de la compañía.

1. Reserva RBNS: Cantidad estimada para siniestros notificados, pero no liquidados completamente. Se calcula restando los pagos parciales ya realizados al coste último estimado del siniestro.

$$\text{RBNS} = \text{Coste último estimado} - \text{Pagos parciales realizados}$$

2. Reserva IBNR: Cantidad estimada para siniestros ocurridos, pero no notificados. Se estima usando métodos actuariales.
3. Reserva de gastos de gestión: Incluye costes administrativos para gestionar siniestros. Suele calcularse como un porcentaje de las reservas RBNS e IBNR, basado en datos históricos.

Por lo tanto, la fórmula de la Reserva Total que se corresponde con los pagos futuros totales sin tener en cuenta los tipos de interés es:

$$\text{Reserva Total} = \text{RBNS} + \text{IBNR} + \text{Gastos de Gestión}$$

2.1. CÁLCULO DE RESERVAS CON MÉTODOS ACTUARIALES TRADICIONALES

A lo largo del tiempo, la ciencia actuarial ha trabajado con métodos que se basan principalmente en el uso de datos agregados y en ciertos supuestos sobre cómo evolucionan los siniestros con el paso del tiempo. Aunque estas técnicas han sido muy útiles y se siguen utilizando ampliamente en el sector asegurador, también tienen sus limitaciones, especialmente cuando se aplican a conjuntos de datos más complejos, con mucha variabilidad o donde las relaciones entre variables no son lineales.

2.1.1. TRIÁNGULOS DE DESARROLLO

El punto de partida de los métodos tradicionales es el uso de triángulos de desarrollo o *run-off triangles*. Estos triángulos recogen información agregada sobre siniestros clasificados por año de ocurrencia (fila) y año de desarrollo (columna), es decir, el número de años transcurridos desde la ocurrencia. A partir de esta estructura se modela el desarrollo esperado de los pagos o del número de siniestros aún abiertos (Wüthrich y Merz, 2008). Formalmente, se denota como:

- i: el año de ocurrencia (accident year)
- j: el año de desarrollo (development year)
- $C_{i,j}$: la cantidad acumulada pagada o número acumulado de siniestros hasta el año de desarrollo j para los siniestros ocurridos en i.

Entonces el triángulo está compuesto por los valores $C_{i,j}$ para los que $j \leq n - i + 1$, donde k es el número total de años observados.

		AÑO DE DESARROLLO				
		Año 0	**Año 1**	**Año 2**	**Año 3**	**Año 4**
AÑO DE OCURRENCIA	**Año 0**	$C_{0,0}$	$C_{0,1}$	$C_{0,2}$	$C_{0,3}$	$C_{0,4}$
	Año 1	$C_{1,0}$	$C_{1,1}$	$C_{1,2}$	$C_{1,3}$	-
	Año 2	$C_{2,0}$	$C_{2,1}$	$C_{2,2}$	-	-
	Año 3	$C_{3,0}$	$C_{3,1}$	-	-	-
	Año 4	$C_{4,0}$	-	-	-	-

Tabla 1. Ejemplo de Triangulo de desarrollo para 5 años. Elaboración propia.

El objetivo de los modelos tradicionales es estimar los valores faltantes del triángulo, es decir, los pagos futuros, y así calcular la reserva como:

$$Reserva = \sum_{i=0}^{k} \sum_{j=k-i+2}^{k} \hat{C}_{i,j}$$

Donde $\widehat{C}_{i,j}$ son los valores estimados.

Esta forma de organizar los datos permite la aplicación de diversos modelos.

2.1.2. MODELOS TRADICIOINALES PARA CÁCULO DE RESERVAS

Existen diversos modelos clásicos para el cálculo de las reservas. En esta sección, se explicarán brevemente los más comunes.

2.1.2.1. MÉTODO CHAIN-LADDER

El método CL es uno de los más conocidos y usados en la práctica actuarial. Es un método que estima los pagos futuros aplicando factores de desarrollo, que se calculan como promedios de razones históricas entre años consecutivos. La suposición básica es que el patrón pasado del desarrollo es representativo del comportamiento futuro. Es decir, este método ayuda a estimar cuánto dinero se tendrá que pagar en el futuro, basándose en cómo se han comportado los pagos en el pasado.

La idea principal es sencilla: se observa cómo han ido creciendo los pagos de un año a otro en el pasado y se calcula un promedio de ese crecimiento. Después, se asume que en el futuro los pagos crecerán de manera parecida. Es decir, se parte de la base de que el comportamiento del pasado es una buena guía para lo que ocurrirá en el futuro.

El método CL se puede aplicar de varias maneras. La primera es una versión más simple determinista), que calcula los importes esperados sin tener en cuenta la incertidumbre, tal y como hemos explicado. Una de las versiones estocásticas de CL, fue propuesta por Mack (Mack, 1993). El modelo de Mack añade una estimación de la variabilidad o "margen de error" en los cálculos, lo que permite no solo dar una cifra estimada, sino también un rango dentro del cual es probable que se encuentre el valor real. Esto es muy útil porque ayuda a las aseguradoras a prepararse mejor ante posibles desviaciones en los pagos futuros.

El modelo CL se expresa mediante:

$$\widehat{f_j} = \frac{\sum_{i=0}^{k-j} C_{i,j+1}}{\sum_{i=0}^{k-j} C_{i,j}}, j = 0,1,2,\dots,k-1$$

Donde $\widehat{f_j}$ son los factores de desarrollo promedio entre los años *j* y *j + 1*.

Los valores futuros se estiman de forma recursiva.

$$\hat{C}_{i,j+1} = \hat{C}_{i,j} \cdot \widehat{f_j}$$

Este modelo puede usarse en versión acumulada o incremental, y sus principales hipótesis son:

- Los factores de desarrollo son constantes en el tiempo.
- Los pagos de distintos años de ocurrencia son independientes.
- No hay cambios estructurales.

El modelo de Mack aporta una extensión estocástica del CL, como se ha mencionado previamente, al introducir una estructura de varianza en los errores. Bajo este modelo:

$$E\left[C_{i,j+1}\middle|C_{i,j}\right] = f_j \cdot C_{i,j}$$

$$Var\left[C_{i,j+1}\middle|C_{i,j}\right] = \sigma_j^2 \cdot C_{i,j}^2$$

Este modelo asume independencia entre las filas del triángulo, pero permite modelar la variabilidad de los pagos futuros y calcular intervalos de confianza para la reserva total y por grupo.

2.1.2.2. MODELOS LINEALES GENERALIZADOS

Los GLM ofrecen una generalización de los métodos anteriores bajo una formulación estadística rigurosa. En este caso, los pagos se modelan como una función del año de desarrollo del siniestro y del año de ocurrencia, bajo la hipótesis de una distribución generalmente de Poisson o Gamma, dependiendo del tipo de variable y la dispersión observada (Wüthrich y Merz, 2008).

Se modela el valor incremental $Y_{i,j}$ como:

$$[Y_{i,j} \sim Distribución\ Exponencial, \qquad log\left(E\left[Y_{i,j}\right]\right) = \mu + \alpha_i + \beta_j]$$

Donde:

- μ : Término independiente.
- α_i: Efecto del año de ocurrencia.
- β_j: Efecto del año de desarrollo.

Los GLM permiten mayor flexibilidad que los métodos anteriores, así como la inclusión de covariables. En este sentido, en este trabajo, lo utilizamos para la inclusión de predictores adicionales y para el tratamiento siniestro a siniestro de forma granular junto con técnicas de ML. Además, es un método estocástico que genera el caso CL al utilizar la distribución de Poisson, por ello se pueden calcular medidas de error como ocurre con el modelo de Mack.

2.1.3. LIMITACIONES DE LOS MÉTODOS CLÁSICOS

Aunque las metodologías tradicionales ofrecen una base sólida para la estimación de reservas, presentan algunas debilidades fundamentales cuando se aplican a contextos más complejos (Lozano y González, 2010):

- Pérdida de información: Los métodos tradicionales trabajan con datos agregados por cohortes anuales, lo que implica una pérdida significativa de la granularidad original de los siniestros individuales. Este enfoque no permite explotar variables individuales como la gravedad del daño, el tipo de accidente o la duración de las hospitalizaciones.
- Supuestos restrictivos: Muchos modelos clásicos suponen independencia entre los desarrollos, linealidad en las relaciones o distribuciones de errores específicas (Normal, Poisson, Gamma), lo cual puede no cumplirse en la práctica.

- Dificultad ante no linealidades e interacciones: Los métodos agregados no capturan adecuadamente interacciones complejas entre variables ni patrones no lineales en los datos.

2.2. CÁLCULO DE RESERVAS CON TÉCNICAS DE MACHINE LEARNING

En la última década, el auge de la ciencia de datos y del aprendizaje automático (*Machine Learning*, ML) ha revolucionado la forma en que se abordan los problemas predictivos en numerosos sectores, incluido el actuarial. A diferencia de los métodos tradicionales, las técnicas de ML trabajan con datos a nivel individual y se centran en la predicción a partir de un entrenamiento supervisado sobre datos históricos.

2.2.1. FUNDAMENTOS DEL ENFOQUE DE ML

Un problema típico de ML supervisado consiste en encontrar una función *f*: $R^p \rightarrow R$ que relacione un vector de variables explicativas x con una variable objetivo y, minimizando una función de perdida $L(y,\hat{y})$, donde $\hat{y} = f(x)$.

Por ejemplo:

$$min_f \frac{1}{k} \sum_{i=0}^{k} L(y_i, f(x_i))$$

Las funciones de pérdida más comunes son el error cuadrático medio (en inglés *Mean Square Error,* MSE) y el error absoluto medio (en inglés *Mean Absolut Error*, MAE). La función *f* puede representar un modelo lineal, una red neuronal, un conjunto de árboles, entre otros (Datacamp, 2024).

El enfoque de ML se basa en tres etapas fundamentales:

1. Preprocesamiento de los datos: Se incluyen tareas como la limpieza de datos, la imputación de valores faltantes, la normalización, la codificación de variables categóricas y la creación de nuevas variables derivadas. A diferencia de los métodos tradicionales, en ML se pueden incorporar múltiples fuentes de información y atributos heterogéneos.

2. Entrenamiento del modelo: El conjunto de datos se divide generalmente en subconjuntos de entrenamiento y validación. Se entrena el modelo para minimizar una función de pérdida, como el error cuadrático medio. Se utilizan técnicas como validación cruzada y regularización para evitar el sobreajuste (*overfitting*).

3. Evaluación y aplicación: Se evalúa el modelo con métricas como RMSE (*Root Mean Squared Error*), MAE o RNMSE (*Root Normalized Mean Squared Error)*, y se implementa para predecir los pagos pendientes o la reserva total.

2.2.3. MODELOS DE MACHINE LEARNING APLICABLES

Existen múltiples algoritmos y arquitecturas que pueden aplicarse al problema de estimación de reservas. Entre los más destacados se encuentran los árboles de decisión, los modelos penalizados, las redes neuronales y los modelos k vecinos más cercanos.

2.2.3.1. MODELOS DE ÁRBOLES DE DECISIÓN

En el contexto actuarial, especialmente en la estimación de pagos futuros asociados a reservas técnicas, los modelos de árboles de decisión han demostrado ser herramientas eficaces debido a su capacidad para modelizar relaciones no lineales y manejar tanto variables continuas como categóricas sin necesidad de transformaciones previas. Este uso ha sido demostrado en el estudio de De Felice y Moriconi (2019), donde se utilizan árboles de decisión para la reserva de siniestros individuales, mostrando mejoras en la precisión y la interpretación frente a métodos tradicionales. Estos modelos consisten en estructuras jerárquicas que segmentan el espacio de datos en subconjuntos homogéneos mediante divisiones sucesivas basadas en los valores de las variables explicativas, buscando minimizar la heterogeneidad dentro de cada nodo terminal (Hastie et al., 2009).

El objetivo principal de los árboles de decisión en este contexto es generar predicciones precisas sobre variables clave como el importe del siniestro o el tiempo hasta su liquidación, permitiendo así una mejor estimación de los flujos de caja futuros. Además, estos modelos pueden utilizarse como bloques de construcción para algoritmos más sofisticados como *Random Forest* y *Gradient Boosting* (por ejemplo, *XGBoost* o *LightGBM*), ampliamente utilizados por su elevada capacidad predictiva y robustez.

Random Forest, introducido por Breiman (2001), combina múltiples árboles construidos a partir de muestras aleatorias con reemplazo del conjunto de entrenamiento (*bootstrap*) y mediante la aleatorización de las variables seleccionadas en cada división. La predicción final se obtiene promediando los resultados de todos los árboles individuales, lo que reduce significativamente la varianza del modelo y lo hace menos susceptible al sobreajuste.

Por otro lado, el método de *Gradient Boosting* construye árboles de forma secuencial, de modo que cada nuevo árbol intenta corregir los errores cometidos por el conjunto anterior, minimizando una función de pérdida diferenciable. Esta estrategia le otorga una gran precisión en tareas de regresión, como la estimación del coste acumulado de siniestros, aunque requiere una cuidadosa regularización para evitar el sobreajuste.

2.2.3.2. MODELOS LINEALES GENERALIZADOS

Los GLM, introducidos por Nelder y Wedderburn (1972), constituyen una extensión del modelo lineal clásico que permite modelar variables respuesta cuya distribución pertenece a la familia exponencial, adecuándose así a muchas de las situaciones típicas en el ámbito actuarial. En el contexto de ML, los GLM representan una de las aproximaciones supervisadas más utilizadas cuando se desea mantener una estructura interpretable, especialmente útil en la predicción de pagos futuros asociados a siniestros.

La formulación de un GLM se basa en tres elementos: una distribución para la variable respuest, una función de enlace que relaciona la media de la variable respuesta con una combinación lineal de las covariables, y un conjunto de predictores que describen las

características del siniestro o del asegurado. Esta flexibilidad convierte a los GLM en herramientas idóneas para el cálculo de reservas en seguros de no vida a partir de información individual, lo que se conoce como *reserving micro-level*.

La estructura de los GLM permite modelar directamente la esperanza matemática del importe de un pago pendiente, condicionada a un vector de características explicativas como la antigüedad del siniestro, tipo de cobertura, edad del asegurado, canal de contratación, entre otras. Esto resulta fundamental en la estimación individualizada de reservas técnicas, permitiendo abandonar los métodos agregados tradicionales (como CL) y migrar hacia un enfoque granular y más preciso (Wüthrich y Merz, 2008).

En el contexto de ML, los GLM pueden ser considerados como modelos base que, aunque paramétricos y de forma funcional específica, sirven de referencia por su balance entre interpretabilidad y rendimiento. Su implementación en entornos computacionales (como en R usando la función glm del paquete stats) permite una calibración eficiente incluso sobre grandes volúmenes de datos históricos.

2.2.3.3. MODELOS PENALIZADOS: LASSO

Los modelos penalizados, y en particular el modelo LASSO (*Least Absolute Shrinkage and Selection Operator*), representan una extensión natural de los GLM en contextos donde se requiere seleccionar automáticamente variables explicativas relevantes y mejorar la capacidad predictiva del modelo. Introducido por Tibshirani (1996), el LASSO incorpora una penalización L1 sobre la suma absoluta de los coeficientes, lo que induce una regularización del modelo y la posibilidad de forzar a cero algunos coeficientes, facilitando así la selección automática de variables.

Desde el punto de vista actuarial, y en particular en el cálculo de reservas, el LASSO puede emplearse para modelizar el importe esperado de pagos futuros (reserva pendiente) a partir de un amplio conjunto de covariables asociadas al siniestro: edad del asegurado, tiempo desde la ocurrencia, clase de siniestro, región geográfica, etc. Su capacidad para manejar problemas de alta dimensionalidad lo hace especialmente útil en bases de datos modernas donde el número de variables explicativas puede ser elevado y algunas de ellas estar altamente correlacionadas.

Los dos beneficios principales del uso de penalizaciones como la del LASSO en este tipo de modelos son:

1. Selección automática de variables: al aplicar la penalización L1, el modelo tiende a reducir los coeficientes de las variables menos relevantes hasta hacerlos cero. Esto permite identificar las covariables más informativas en la predicción del importe de reserva, lo cual mejora la interpretabilidad del modelo y reduce el riesgo de sobreajuste.

2. Control de la multicolinealidad: en presencia de variables correlacionadas, los modelos lineales tradicionales presentan inestabilidad en la estimación de los coeficientes. La penalización L1 introduce un sesgo controlado que, a cambio, disminuye la varianza de las estimaciones, favoreciendo así una mayor robustez del modelo (Hastie et al., 2009).

En la práctica, el LASSO puede integrarse en marcos de regresión de siniestros para predecir el valor esperado de los pagos pendientes a nivel individual, utilizando como base histórica tanto características del asegurado como del siniestro. Esta aproximación ofrece una alternativa flexible a los métodos tradicionales basados en triángulos agregados, con la ventaja adicional de incorporar variables externas y covariables temporales.

$$LASSO: min_{\beta}\left(\sum_{i=0}^{k}(y_i - X_i^{\top}\beta)^2 + \lambda\sum_{j=0}^{p}|\beta_j|\right)$$

2.2.3.4. MODELOS K VECINOS MÁS CERCANOS

El modelo de los K vecinos más cercanos (KNN, por sus siglas en inglés: *K-Nearest Neighbors*) constituye una técnica no paramétrica ampliamente utilizada en ML para resolver tanto problemas de clasificación como de regresión (Cover y Hart, 1967). Su lógica se fundamenta en el supuesto de que observaciones similares tienden a producir resultados similares, principio que resulta especialmente útil en contextos con estructuras de datos complejas o sin supuestos funcionales claros.

En su aplicación al ámbito actuarial, concretamente en la estimación de pagos futuros asociados a siniestros individuales, el algoritmo KNN permite predecir el importe esperado de un siniestro no liquidado a partir del comportamiento histórico de otros siniestros con características similares. Para ello, el modelo identifica los K siniestros más próximos en el espacio de características utilizando una medida de distancia, como la Euclidea, Manhattan o de Mahalanobis.

En problemas de regresión —caso habitual en la predicción de pagos pendientes— la predicción para un nuevo siniestro corresponde a la media (o mediana, en versiones más robustas) del valor objetivo observado en sus K vecinos más cercanos. De esta manera, el modelo genera una estimación del valor esperado del pago futuro sin necesidad de asumir una distribución a priori para la variable respuesta.

Una de las principales ventajas del método KNN radica en su simplicidad conceptual y en su capacidad para adaptarse a relaciones no lineales entre las variables. No obstante, presenta ciertas limitaciones: la elección del parámetro K es crítica para su rendimiento. Valores bajos pueden conducir a una alta varianza y sensibilidad a *outliers*, mientras que valores elevados tienden a introducir sesgo al considerar observaciones menos representativas. Asimismo, el método puede ser computacionalmente costoso en bases de datos extensas, aunque este problema puede mitigarse mediante técnicas de indexación espacial como *KD-Trees o Ball Trees* (Hastie et al., 2009).

En el contexto de cálculo de reservas individuales, el modelo KNN puede servir como una herramienta complementaria a los enfoques clásicos y paramétricos, ofreciendo predicciones razonables incluso cuando las relaciones funcionales entre las covariables y la variable respuesta no están claramente definidas. Aunque menos frecuente en entornos regulados donde la interpretabilidad es clave, su utilidad aumenta cuando se prioriza la precisión predictiva y se dispone de grandes volúmenes de datos etiquetados.

2.3. VENTAJAS Y DESAFÍOS

El uso de técnicas de ML en el cálculo de reservas presenta una serie de ventajas que explican el creciente interés que han despertado en el ámbito actuarial. Una de las principales fortalezas de estos modelos es su capacidad para detectar patrones complejos y no evidentes en los datos. A diferencia de los métodos tradicionales, que suelen basarse en relaciones lineales o estructuras predefinidas, los algoritmos de aprendizaje automático pueden identificar interacciones entre variables y comportamientos no lineales que serían difíciles de captar con enfoques clásicos.

Otra ventaja relevante es la mejora en la precisión predictiva. Al trabajar directamente con datos individuales y manejar un mayor volumen de información, los modelos de ML suelen ofrecer estimaciones más ajustadas a la realidad, lo que puede traducirse en una mayor fiabilidad a la hora de establecer reservas. Además, estos modelos pueden actualizarse de manera continua con la incorporación de nuevas observaciones, lo que facilita su automatización y adaptación a cambios en las características de la cartera o del entorno.

Un aspecto adicional que destacar es la posibilidad de interpretar las predicciones a través de herramientas específicas desarrolladas para este fin. Técnicas como los *SHAP values* (Lundberg y Lee, 2017) cuantificar el peso que cada variable tiene en la predicción final del modelo, lo que mejora la transparencia y facilita la comprensión del comportamiento del modelo, algo especialmente valioso en contextos donde se requiere justificar las decisiones ante auditores o supervisores.

No obstante, el uso de ML también conlleva una serie de desafíos importantes. En primer lugar, estos modelos requieren bases de datos completas, detalladas y correctamente estructuradas. La calidad y la granularidad de los datos son factores determinantes para el éxito del proceso predictivo. Además, el entrenamiento y validación de los modelos exige una carga computacional mayor que la de los métodos actuariales tradicionales, así como una supervisión más rigurosa para evitar problemas de sobreajuste, es decir, que el modelo funcione muy bien sobre los datos de entrenamiento, pero tenga un bajo rendimiento con datos nuevos.

Por último, es importante señalar que la normativa contable y regulatoria en el sector asegurador continúa fundamentándose, en gran medida, en modelos tradicionales por su simplicidad y facilidad de justificación. Esto puede suponer una barrera para la adopción de modelos más complejos, incluso si estos ofrecen mejores resultados en términos técnicos.

2.4. APORTACIÓN A LOS OBJETIVOS DE DESARROLLO SOSTENIBLE

El presente Trabajo Fin de Máster se alinea con varios Objetivos de Desarrollo Sostenible (ODS), especialmente en lo relativo a la promoción de la innovación, la eficiencia en la gestión de recursos y el fortalecimiento institucional del sector financiero. En particular, contribuye al ODS 9 (Industria, Innovación e Infraestructura), al aplicar técnicas avanzadas de *Machine L*earning al cálculo de reservas técnicas, un proceso crítico dentro del ámbito actuarial y asegurador. Esta innovación metodológica favorece la modernización de la industria mediante la incorporación de herramientas analíticas más precisas y adaptativas.

Asimismo, se relaciona con el ODS 12 (Producción y Consumo Responsables), en tanto que mejora la estimación de los pasivos futuros de las aseguradoras permite una gestión más responsable, eficiente y sostenible de sus recursos financieros. Esta optimización contribuye también al ODS 8 (Trabajo Decente y Crecimiento Económico), al impulsar la productividad de los procesos actuariales, fomentar nuevas competencias profesionales basadas en el análisis de datos, y fortalecer la resiliencia económica del sector. Por último, si bien en menor medida, el proyecto se alinea con el ODS 17 (Alianzas para lograr los objetivos), al promover la colaboración entre el ámbito académico y el sector asegurador en la búsqueda de soluciones innovadoras con impacto práctico.

3. MARCO PRÁCTICO

Con el objetivo de explorar el potencial del aprendizaje automático en el ámbito del cálculo de reservas técnicas, el presente trabajo incluye una aproximación práctica basada en la simulación de datos realistas y el entrenamiento de modelos predictivos, así como la comparación de la capacidad predictiva de distintas técnicas de ML. El foco principal se sitúa en la estimación de los pagos futuros asociados a una cartera de seguros, con el propósito de evaluar su aplicabilidad práctica en el contexto actuarial. Dado que no se dispone de una base de datos real que cumpla con los requisitos técnicos y de granularidad necesarios para este tipo de análisis, se ha optado por el uso de un simulador especializado, el cual permite generar datos sintéticos representativos del comportamiento de siniestralidad de una cartera de seguros de responsabilidad civil de automóviles.

La simulación se centrará específicamente en siniestros por lesiones corporales derivados de accidentes de tráfico, en el marco de una cartera ficticia con nueve años de antigüedad. Este tipo de siniestros es especialmente interesante desde el punto de vista actuarial, ya que combina baja frecuencia con alta severidad, genera pagos fraccionados a lo largo del tiempo y presenta una elevada incertidumbre en los plazos de reporte y liquidación. Estas características lo convierten en un entorno idóneo para poner a prueba la capacidad de los modelos de ML para mejorar la estimación de reservas frente a los métodos clásicos.

El análisis práctico de este trabajo se estructura del siguiente modo. En primer lugar, se describe el proceso de generación de datos sintéticos mediante el simulador *SynthETIC* (Avanzi et al., 2021), explicando las principales características de la base de datos resultante y su adecuación al estudio del comportamiento de una cartera de seguros de automóviles. A continuación, se aplican cuatro técnicas de aprendizaje automático sobre estos datos simulados. Para cada modelo, se realiza un proceso completo de entrenamiento, validación y evaluación, no solo con el objetivo de identificar su capacidad predictiva, sino también de utilizarlos para la estimación directa de pagos futuros. En particular, cada modelo se emplea para calcular las reservas asociadas a los siniestros declarados, pero no completamente pagados (RBNS), obteniéndose así una estimación completa de la reserva total. Finalmente, se comparan los resultados de las técnicas de ML con los obtenidos mediante métodos actuariales tradicionales, con el fin de evaluar sus diferencias en términos de precisión, estabilidad y aplicabilidad operativa. El punto de partida de este marco práctico es, por tanto, la creación de una base de datos artificial suficientemente representativa sobre la cual puedan implementarse y contrastarse distintas metodologías de cálculo de reservas.

3.1. GENERACIÓN DE DATOS CON EL SIMULADOR *SynthETIC*

Dado que en el sector asegurador no es habitual contar con bases de datos individuales de siniestros abiertas al público, y menos aún con la granularidad y trazabilidad necesarias para evaluar el cálculo de reservas a nivel individual, este trabajo recurre a la generación de datos sintéticos. Para ello se utiliza el simulador *SynthETIC*, un paquete desarrollado en el entorno R específicamente orientado a seguros de no vida. Su objetivo es replicar el comportamiento estadístico de carteras reales de siniestros, permitiendo así analizar y comparar distintas metodologías de estimación de reservas sin comprometer datos confidenciales.

El simulador *SynthETIC* fue desarrollado inicialmente por Avanzi et al. (2021), como se expone en su trabajo *SynthETIC*: *An individual insurance claim simulator with feature control*. Posteriormente, Wang, Wüthrich y el grupo de trabajo de Data Science de la Asociación Suiza de Actuarios (SAV) adaptaron y ampliaron esta herramienta mediante el desarrollo del algoritmo *data simulation.R*, descrito en I*ndividual claims generator for claims reserving studies: data simulation.R* (Wang et al., 2022). Este script incorpora una estructura modular sobre el paquete original *SynthETIC*, permitiendo la generación de bases de datos sintéticas más representativas y ajustadas a la realidad de carteras de seguros.

3.1.1. DESCRIPCIÓN TÉCNICA DE LA BASE DE DATOS

El simulador se ha calibrado para recrear una cartera de seguros de responsabilidad civil de vehículos por lesiones corporales, con una antigüedad inicial de diez años (1 de enero de 2013 y el 31 de diciembre de 2021). En un primer momento, se generaron dos bases de datos: una de siniestros con 61.307 registros y otra de pagos con 87.441 observaciones. Sin embargo, al integrar ambas fuentes a través del identificador único de siniestro (Id), se identificaron inconsistencias en los datos correspondientes al ejercicio 2012. Por este motivo, se decidió acotar la ventana temporal de análisis al período comprendido entre el 1 de enero de 2013 y el 31 de diciembre de 2021, descartando los registros del año inicial.

Además, para el análisis predictivo y el cálculo de reservas se eliminaron los siniestros con estado IBNR, ya que su naturaleza (no reportados) impide la validación empírica de las predicciones en términos de pagos reales. Esta depuración permite trabajar sobre un conjunto de datos más robusto y realista, en el que se dispone de la variable objetivo-observada para los siniestros cerrados y RBNS, lo cual facilita una evaluación directa de la calidad de los modelos predictivos aplicados.

Desde un punto de vista técnico, el modelo parte de la simulación de la frecuencia de siniestros mediante una distribución de Poisson, mientras que la severidad, es decir, el importe económico de los siniestros se modela con una distribución *Power-normal*, que permite capturar tanto la asimetría como la existencia de colas pesadas. Además, se incorpora una tasa de inflación del 2% anual, lo que añade realismo financiero al proceso de simulación, representando el valor temporal del dinero y su impacto en la magnitud final de los pagos.

El algoritmo *datasimulation.R* añade funcionalidades que potencian aún más el conjunto de datos. Por ejemplo, incorpora variables como la edad del lesionado o el tipo de siniestro, factores que se sabe que influyen en la gravedad del caso, los tiempos de tramitación y los importes finales.

El resultado de la simulación es una base de datos individualizada, compuesta por dos archivos principales:

- SINIESTROS (siniestros.csv)

La base de datos de siniestros individuales, donde cada fila representa un siniestro y contiene 15 campos clave, entre ellos:

	CAMPO	DESCRIPCIÓN
1	Id	Identificador único del siniestro
2	Type	Tipo de siniestro (con 6 categorías posibles, del 1 al 6)
3	Age	Edad del lesionado (entre 18 y 65 años)
4	AccDate	Fecha de ocurrencia del siniestro (yyyy/mm/dd)
5	AccMonth	Número de meses desde la ocurrencia del siniestro (de 1 a 108 meses)
6	AccWeekday	Día de la semana en que ocurrió el siniestro (de lunes a domingo)
7	RepDate	Fecha de reporte del siniestro (yyyy/mm/dd)
8	RepMonth	Fecha de reporte del siniestro (en unidades mensuales) desde 1 hasta 108 meses)
9	RepDelDays	Retraso en el reporte del siniestro (en días)
10	SetMonth	Fecha de liquidación (en unidades mensuales) desde 1 hasta 108
11	SetDelMonths	Retraso en la liquidación (en unidades mensuales)
12	Ultimate	Coste total estimado
13	PayCount	Número total de pagos realizados
14	Status	Estado del siniestro (Cerrados, RBNS)
15	CumPaid	Pagos acumulados hasta el corte de datos (para siniestros Cerrados y RBNS)

Tabla 2. Tabla descriptiva de los campos de los siniestros individuales. Elaboración propia.

Se crearon, además, varias variables, a partir de las originales, para alcanzar los resultados deseados.

	CAMPO	DESCRIPCIÓN
1	AccYearMonth	Mes de ocurrencia del siniestro cerrado (de 1 a 12).
2	RepYearMonth	Mes de reporte del siniestro cerrado (de 1 a 12)
3	SetDelDays	Días de retraso en la liquidación del siniestro desde la fecha de reporte.
4	RepDelMonths	Meses de retraso en el reporte del siniestro desde la ocurrencia de dicho siniestro.
5	AccYear	Año de ocurrencia del siniestro
6	RepYear	Año de reporte del siniestro

Tabla 3. Tabla descriptiva de los campos adicionales. Elaboración propia

- PAGOS (pagos.csv)

La base de datos de pagos recoge el detalle de cada evento asociado a los siniestros, ya sea un pago o una actualización de su estado. Esta información resulta fundamental para aplicar métodos estocásticos de cálculos de reservas, ya que dichos métodos requieren conocer el desarrollo de los siniestros a través de sus pagos parciales. La base incluye las siguientes variables:

	CAMPO	DESCRIPCIÓN
1	Id	Identificador del siniestro
2	EventId	Identificador del evento
3	EventMonth	Mes en el que ocurre el evento
4	Paid	Importe del pago parcial asociado
5	PayInd	Identificador del pago
6	OpenInd	Indicador binario que refleja si el siniestro continúa abierto (1) o ha sido cerrado (0) en fecha de 31/12/2021

Tabla 4. Tabla descriptiva de los campos de los pagos de cada evento. Elaboración propia.

Este diseño permite trabajar con enfoques tanto agregados (como CL, que puede construirse a partir de los datos acumulados) como individuales (como los modelos

de ML, que se benefician de la trazabilidad completa del proceso de cada siniestro). Además, disponer de información sobre el estado del siniestro y sobre pagos parciales permite explorar la predicción tanto del coste total como del desarrollo futuro de cada caso, lo cual es esencial para tareas de cálculo de reservas.

En cuanto a volumen, tras el proceso de depuración y unión de las bases de datos de siniestros y pagos (eliminando registros con inconsistencias y restringiendo el periodo de análisis a 2013–2021), se obtienen en torno a 77.099 registros de pagos parciales. Esta cantidad garantiza una base de datos suficientemente extensa para realizar entrenamientos robustos y particionar los datos en conjuntos de entrenamiento, validación y prueba, asegurando la calidad estadística de los modelos y la posibilidad de generalizar los resultados. Además, este volumen de información resulta clave para el posterior cálculo de los pagos futuros esperados, permitiendo así estimar la reserva total asociada a la cartera simulada mediante técnicas de aprendizaje automático.

3.1.2. ANÁLISIS ESTADÍSTICO DE LA BASE DE DATOS

En la sección que sigue se presenta el proceso de elaboración de modelos predictivos basados en técnicas de ML. Como punto de partida, es fundamental identificar y seleccionar las variables relevantes, así como realizar un análisis estadístico preliminar que nos permita entender mejor los datos. El análisis se ha llevado a cabo utilizando la base de datos "panel5.csv", donde cada fila representa un pago parcial por año de ocurrencia y de desarrollo distinto.

Dado que el objetivo del trabajo es estimar los pagos futuros de siniestros ya reportados (cerrados y RBNS), se emplea como variable objetivo la variable creada *Paid*, que representa el importe pagado de forma parcial en cada combinación de año de ocurrencia y año de desarrollo. Esta variable ha sido construida a partir de la base de datos de pagos, agrupando los registros por siniestro (*Id*), año de ocurrencia y año de pago, y unida posteriormente con la base de siniestros mediante la variable común Id.

La base de datos utilizada para el modelado es *panel5*, una estructura en formato panel que permite capturar la evolución temporal de los pagos por siniestro a lo largo de los distintos años de ocurrencia y de desarrollo. Esta base integra tanto información estática del siniestro como variables acumuladas y temporales, manteniendo únicamente aquellas disponibles hasta la fecha de corte (31/12/2021), lo que simula un entorno real de predicción sin incorporar información futura.

Con esta configuración, el entrenamiento, validación y evaluación de los modelos se realiza sobre siniestros ya conocidos, utilizando como entradas únicamente variables disponibles hasta la fecha de corte, y como objetivo el importe pagado en el año siguiente. Esta estructura en formato panel permite estimar de manera secuencial los pagos futuros para cada siniestro RBNS y, posteriormente, agregarlos para obtener una estimación global de la reserva. A continuación, se realiza un análisis estadístico de la base de datos panel5, con el fin de identificar las variables que pueden tener mayor relevancia en la predicción de *Paid*.

- *Paid* (Variable Objetivo)

  La variable objetivo *Paid* en la base de datos representa el importe de cada pago parcial realizado en un siniestro, desglosado por año de origen (*AccYear*) y año de

desarrollo (*DevYear*). Es decir, para cada combinación de año de ocurrencia del siniestro y su posterior evolución temporal, se recoge el importe que efectivamente ha sido pagado en ese periodo específico. En los casos en que no se ha efectuado ningún pago parcial durante un año de desarrollo determinado, la variable toma el valor 0, indicando la ausencia de desembolsos en ese tramo temporal del desarrollo del siniestro.

Año de origen	Nº de registros	Nº de siniestros únicos	Paid total	Mínimo	Máximo
2013	56.079	6.231	53.634,61	0	952,63
2014	55.980	6.220	54.223,43	0	344,76
2015	56.241	6.249	53.008,00	0	264,00
2016	54.810	6.090	54.378,35	0	396,63
2017	54.279	6.031	57.733,74	0	512,92
2018	55.116	6.124	51.333,19	0	476,66
2019	53.316	5.924	40.532,81	0	252,19
2020	51.840	5.760	26.876,04	0	201,21
2021	34.020	3.780	5.701,45	0	107,98
Total	**471.681**	**52.409**	**397.421,61**	**0**	**952,63**

Tabla 5. Tabla descriptiva de la variable Paid agrupada por año de origen de accidente (miles €). Elaboración propia.

La tabla resume los pagos parciales realizados por año de origen de los siniestros en la base de datos. Para cada año entre 2013 y 2021, se muestran el número total de registros (correspondientes a observaciones anuales de pagos por desarrollo), el número de siniestros únicos implicados, el total pagado en miles de euros, y los valores mínimo y máximo registrados de *Paid* en ese año.

Se observa una tendencia decreciente en los pagos a lo largo del tiempo: los años más antiguos, como 2013 a 2017, presentan cuantías totales de pagos más elevados, en torno a los 50.000-57.000 mil euros, con un número de siniestros únicos relativamente constante (alrededor de 6.000 por año). A partir de 2018 y especialmente en 2020 y 2021, se evidencia una disminución tanto en el número de registros como en el total pagado, lo que es coherente con el hecho de que los siniestros más recientes han tenido menos tiempo de desarrollo y, por tanto, menos oportunidades de generar pagos.

El valor mínimo de *Paid* es 0 en todos los años, reflejando que en algunas combinaciones de año de origen y desarrollo no se ha realizado ningún pago parcial. El valor máximo alcanza los 952,63 mil euros en 2013, destacando la existencia de siniestros con pagos elevados en los primeros años. En conjunto, la tabla muestra un total acumulado de 397.421,61 miles de euros en pagos parciales para 52.409 siniestros únicos a lo largo de 471.681 registros.

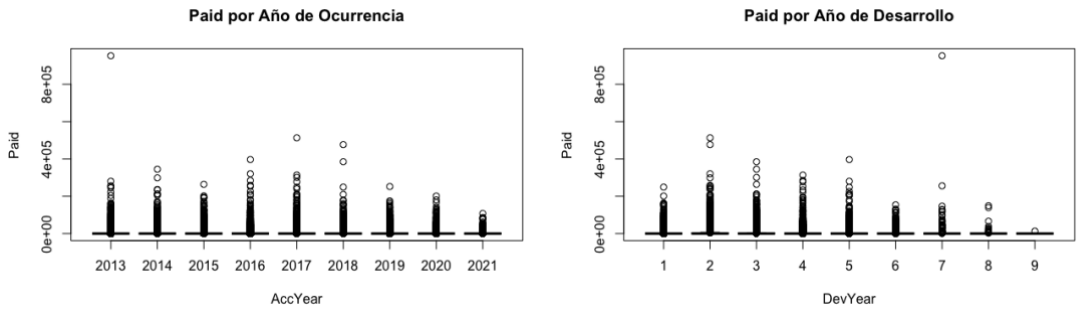

Ilustración 1. Distribución de Paid según año de ocurrencia y de desarrollo

- *Paid vs. Age*

El análisis de Paid en función de la edad muestra una tendencia ligeramente creciente en los valores medios y medianos a medida que aumenta la edad del asegurado. En particular, el grupo de 60 a 69 años presenta la media más alta de pagos acumulados (7.579,74 €), seguido por los grupos de 50 a 59 años (media: 7.043,76 €), y 40 a 49 años (media: 6.773,75 €). En contraste, los grupos más jóvenes, como 18 a 29 años, muestran valores medios menores (media: 2.308 €)

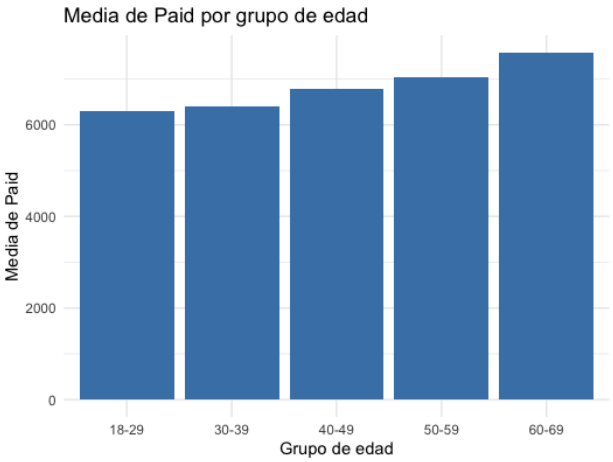

Ilustración 2. Media de Paid por grupo de edad. Elaboración propia.

Estas diferencias se acompañan de una alta dispersión dentro de cada grupo, como lo reflejan las desviaciones estándar (por ejemplo, 17.215 € en el grupo 18–29), lo que sugiere una variabilidad considerable en las cantidades pagadas incluso dentro de los mismos rangos de edad.

En cuanto a las correlaciones, la correlación de Spearman es mayor (0.133), lo que apunta a una relación monótona leve: a mayor edad, tiende a observarse una mayor cantidad, aunque de forma poco consistente.

- *Paid vs. Type*

También se ha examinado la variable *Paid* en función de la variable categórica *Type*, que clasifica los siniestros en seis categorías distintas.

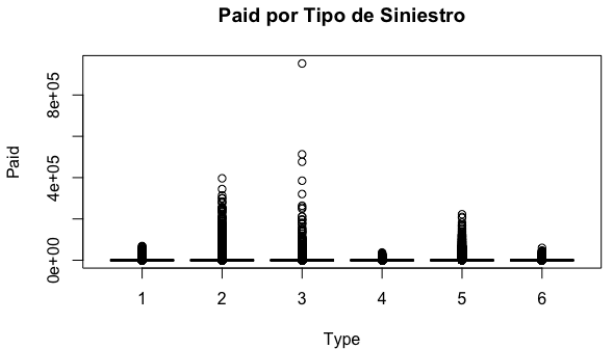

Ilustración 3. Paid por Tipo de siniestro

Los resultados muestran diferencias notables en el promedio de pagos acumulados según el tipo:

El Tipo 2 destaca con un valor medio de 5.066,43 €, muy por encima del resto de los tipos. Le siguen, en un segundo escalón, el Tipo 3 (1.330,15 €) y el Tipo 5 (1.221,17 €). En contraste, los Tipos 1, 4 y 6 presentan valores medios mucho más bajos, todos por debajo de 610 €.

Estas diferencias sugieren que la variable Type tiene un impacto importante en los pagos y podría desempeñar un papel relevante en la segmentación de siniestros para la modelización de reservas.

- *Paid vs. Status*

Estado	Media Paid	Total registros
Cerrado	916,32 €	408.609
RBNS	364,72 €	63.072

Tabla 6. Descriptiva de la variable explicativa Status

Los resultados muestran que, en promedio, los siniestros cerrados presentan valores significativamente más altos de *Paid* en comparación con los siniestros en estado RBNS.

Este comportamiento es coherente con la lógica operativa del ciclo de vida del siniestro, ya que los casos cerrados han completado todo el proceso de indemnización, mientras que los RBNS representan siniestros aún en desarrollo, donde es común que los pagos efectuados hasta la fecha sean parciales o incluso inexistentes.

Desde el punto de vista actuarial, esta variable *Status* resulta especialmente relevante en el contexto del cálculo de reservas. Su inclusión como predictor en modelos de ML puede ayudar a capturar mejor la diferencia estructural entre siniestros ya liquidados y aquellos aún en evolución. Este tipo de segmentación es fundamental para estimar adecuadamente las obligaciones pendientes y mejorar la precisión en la estimación de reservas totales.

3.2. IMPLEMENTACIÓN DE LOS MÉTODOS MACHINE LEARNING

Antes de la aplicación de modelos de ML para la predicción de pagos futuros, se llevó a cabo un exhaustivo proceso de preparación de los datos. En primer lugar, se seleccionaron únicamente aquellas variables con relevancia potencial en el proceso predictivo, descartando aquellas que no aportaban información útil o que podían introducir ruido en los modelos.

A continuación, se excluyeron todas las observaciones que presentaban valores faltantes en la variable objetivo (*Paid*), garantizando así la calidad y completitud de los datos utilizados.

Se realizaron diversas transformaciones sobre variables clave para facilitar su interpretación por parte de los algoritmos. Por ejemplo, variables como *Type*, *Status* y *AccWeekday* fueron transformadas en factores, asegurando una correcta codificación de variables categóricas. Adicionalmente, se filtraron únicamente los siniestros cuyo estado era "*Closed*" o "*RBNS*".

3.2.1. MODELIZACIÓN CON LAS DIFERENTES TÉCNICAS DE MACHINE LEARNNG

En la segunda fase del análisis, se procedió a implementar y evaluar diversos modelos para predecir los pagos futuros. Se aplicaron cuatro técnicas diferentes: *Random Forest, Lasso,* GLM *y* KNN.

Antes de aplicar los modelos, se realizó una partición del conjunto de datos. Se extrajo aleatoriamente una muestra del 15%, de los siniestros disponibles para constituir el conjunto de entrenamiento (*train_sample5*), es decir, 70.752 registros, mientras que el 85% restante (400.929 registros), se utilizó como conjunto de prueba (*test_sample5*). Se utilizó un porcentaje pequeño de entrenamiento debidoal tiempo computacional.

Cada uno de los algoritmos se entrenó utilizando las mismas variables predictoras: *AccYear, DevYear, Type, Age, AccMonth, AccWeekday, RepDelDays y Status*. Tras el entrenamiento, se realizaron predicciones sobre el conjunto de prueba, lo que permitió evaluar la capacidad de generalización de cada modelo frente a datos no vistos durante el entrenamiento.

3.2.1.1. APLICACIÓN DE LAS TÉCNICAS

Una vez entrenados y evaluados los distintos modelos sobre el conjunto de prueba, se recopilaron sus respectivos errores para facilitar la comparación del rendimiento predictivo. En la tabla siguiente se muestran los valores de RMSE y NRMSE obtenidos por cada método:

MÉTODO	RMSE	NRMSE
Random Forest	5.347,11	0,00561
Lasso	5.440,66	0,00571
GLM ML	5.421,55	0,00569
KNN	6.093,41	0,00640

Tabla 7. Errores cometidos con cada método

De los cuatro modelos analizados, GLM obtuvo el menor error absoluto (RMSE), así como el menor error relativo (NRMSE), lo que indica un mejor ajuste general sobre los datos de prueba. Le sigue de cerca el modelo *Random Forest*, con diferencias pequeñas en rendimiento. Por su parte, el modelo KNN presentó el peor desempeño relativo, con un NRMSE sensiblemente más elevado, lo que sugiere menor capacidad para capturar la estructura subyacente en los datos.

Cabe destacar que, si bien las diferencias en RMSE pueden parecer moderadas, la métrica NRMSE, al estar normalizada respecto al rango de la variable objetivo, permite interpretar los resultados de forma más homogénea y comparativa. En este contexto, una mejora de 0,0001 en el NRMSE representa una reducción proporcionalmente relevante del error para la magnitud de pagos analizada.

3.2.1.2. PROYECCIÓN DE PAGOS FUTUROS Y CÁLCULO DE RESERVAS

Una vez completada la fase de validación y comparación de los distintos modelos de aprendizaje automático, se procedió a su aplicación para la estimación de pagos futuros. Esta fase constituye el núcleo del análisis predictivo, ya que permite completar los triángulos de pagos con valores estimados en aquellos años de desarrollo en los que, por razones temporales, no se dispone aún de información real. De esta manera, se pueden proyectar los flujos de salida futuros asociados a siniestros ya ocurridos, lo cual es esencial para el cálculo de reservas técnicas.

El primer paso consistió en identificar, para cada año de ocurrencia (*AccYear*), el máximo año de desarrollo (*DevYear*) en el que aún se observaban pagos (es decir, *Paid* > 0). Esto permitió delimitar de forma precisa cuáles son los años de desarrollo futuros para cada cohorte de siniestros, es decir, aquellos en los que aún no se ha registrado actividad pero que deben ser contemplados en la estimación futura. Esta información se integró al conjunto de datos y se utilizó para filtrar exclusivamente las observaciones correspondientes a los años de desarrollo futuros.

A continuación, cada uno de los modelos previamente entrenados — *Random Forest*, *Lasso*, GLM y KNN — fue utilizado para realizar predicciones sobre estos años futuros. Las predicciones obtenidas corresponden a los pagos acumulados estimados (*Paid_pred*).

Con los resultados de las predicciones, se construyeron distintos triángulos:

- Triángulo real, a partir de los datos históricos disponibles.

AÑO DE ORIGEN	AÑO DE DESARROLLO 1	2	3	4	5	6	7	8	9
2013	5.067,76	18.238,4	14.896,90	8.133,71	4.010,15	1.382,42	1.498,38	393,90	12.97
2014	5.417,13	19.467,94	15.362,71	7.757,76	3.807,48	1.305,98	936,55	167,89	0,00
2015	5.655,37	18.417,87	14.917,33	8.514,53	3.040,80	1.459,03	1.003,08	0,00	0,00
2016	5.423,32	19.528,33	15.049,81	9.094,54	4.084,32	1.198,03	0,00	0,00	0,00
2017	6.071,79	22.252,00	16.232,07	8.470,92	4.706,96	0,00	0,00	0,00	0,00
2018	6.042,22	20.635,23	16.486,92	8.168,82	0,00	0,00	0,00	0,00	0,00
2019	5.314,01	19.464,94	15.753,86	0,00	0,00	0,00	0,00	0,00	0,00
2020	6.384,52	20.491,51	0,00	0,00	0,00	0,00	0,00	0,00	0,00
2021	5.701,45	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00

Tabla 8. Triángulo de desarrollo con los datos históricos

- Triángulo futuro, compuesto únicamente por los pagos predichos.

- Triángulo completo, que combina los pagos reales observados con los pagos futuros estimados, ofreciendo así una visión integral del desarrollo de los siniestros.

Además, se calcularon las versiones acumuladas de los triángulos, lo que permite visualizar la evolución total de los pagos por cohorte a lo largo de los años de desarrollo.

Una vez disponible el triángulo completo, se procedió al cálculo del vector de pagos futuros (VPF), que resume los flujos agregados estimados por año calendario (coincidiendo con la diagonal del triángulo). Este vector constituye la base para el cálculo de la provisión técnica descontada, la cual se obtiene aplicando factores de descuento derivados de la curva de tipos de interés libre de riesgo ajustada por volatilidad, publicada por la Autoridad Europea de Seguros y Pensiones de Jubilación (EIOPA). En este trabajo se ha utilizado la curva publicada a fecha de 5 de mayo de 2025 (ver EIOPA, 2025). Este enfoque permite incorporar el valor temporal del dinero y alinear las estimaciones con criterios actuariales.

La metodología usada parte del supuesto de que los pagos futuros se distribuyen como una renta anual vencida.

Año	1	2	3	4	5	6	7	8
TI libre de riesgo con VA	0,0211	0, 02045	0, 02118	0, 02199	0, 02275	0, 02353	0, 02428	0, 02496

Tabla 9. Tipos de interés libre de riesgo con ajuste por volatilidad a 5 de mayo de 2025 (EIOPA).

Este procedimiento fue implementado de forma sistemática y homogénea para cada uno de los modelos considerados, permitiendo no solo comparar el volumen total de pagos futuros proyectados, sino también evaluar la sensibilidad de la provisión calculada frente a la técnica predictiva empleada.

En los siguientes apartados se detallan los resultados específicos obtenidos con cada técnica de modelización, incluyendo el comportamiento de los pagos estimados, la forma del triángulo generado y el importe final proyectado para las reservas descontadas.

3.2.1.2.1. RESULTADOS CON ÁRBOLES DE DECISION: RANDOM FOREST

El modelo de *Random Forest* se seleccionó por su capacidad para capturar relaciones no lineales y complejas entre las variables explicativas y la variable objetivo. Además, su robustez frente al sobreajuste y su buen desempeño en contextos con muchas variables explicativas lo hacen especialmente útil para problemas de predicción en seguros.

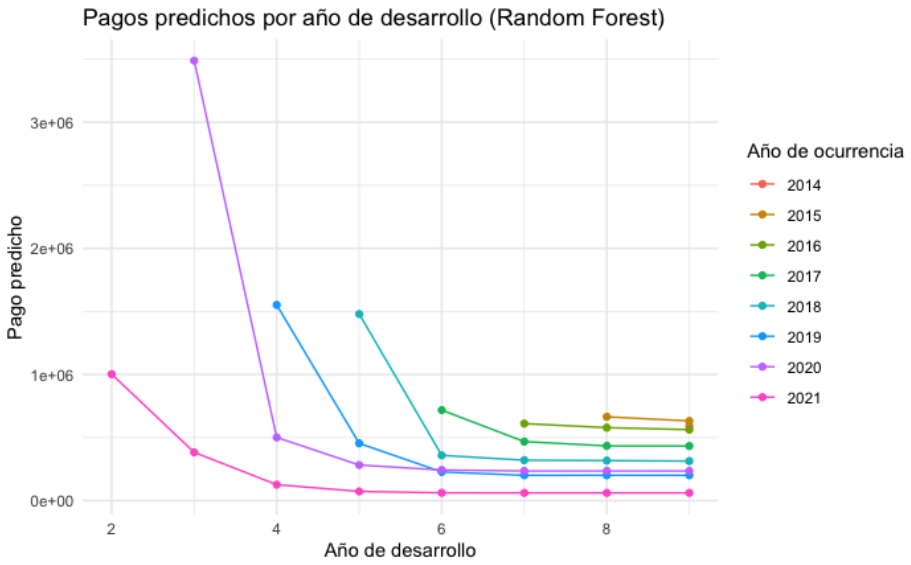

Ilustración 4. Pagos predichos por año de desarrollo con Random Forest. Elaboración propia.

El gráfico muestra un patrón decreciente de pagos predichos por año de desarrollo, coherente con la dinámica típica de los siniestros. Las cohortes más recientes presentan pagos más prolongados en el tiempo, mientras que en las más antiguas solo se predicen valores residuales en los últimos años. En general, el modelo reproduce una forma razonable, aunque con cierta rigidez en las predicciones más lejanas.

A continuación, se construyó el triángulo combinado, integrando los pagos históricos observados con las predicciones generadas por el modelo para años futuros. A partir de este triángulo se obtuvo la versión acumulada, necesaria para evaluar la evolución total de los pagos a lo largo del tiempo.

TRIÁNGULO DE DESARROLLO COMPLETO – RANDOM FOREST

AÑO DE ORIGEN	AÑO DE DESARROLLO 1	2	3	4	5	6	7	8	9
2013	5.067,76	18.238,4	14.896,90	8.133,71	4.010,15	1.382,42	1.498,38	393,90	12.97
2014	5.417,13	19.467,94	15.362,71	7.757,76	3.807,48	1.305,98	936,55	167,89	593,04
2015	5.655,37	18.417,87	14.917,33	8.514,53	3.040,80	1.459,03	1.003,08	664,75	631,87
2016	5.423,32	19.528,33	15.049,81	9.094,54	4.084,32	1.198,03	610,86	578,78	562,91
2017	6.071,79	22.252,00	16.232,07	8.470,92	4.706,96	717,57	468,10	434,15	433,09
2018	6.042,22	20.635,23	16.486,92	8.168,82	1.479,64	359,09	320,74	317,78	313,71
2019	5.314,01	19.464,94	15.753,86	1.551,55	454,41	227,36	201,11	201,05	201,07
2020	6.384,52	20.491,51	3.487,87	501,21	282,57	242,96	235,51	235,51	235,52
2021	5.701,45	100,28	383,07	126,59	73,23	62,12	61,65	61,65	61,67

Tabla 10. Triángulo de desarrollo completo con el método Random Forest (miles €). Elaboración propia.

TRIÁNGULO DE DESARROLLO ACUMULADO – RANDOM FOREST

AÑO DE ORIGEN	AÑO DE DESARROLLO 1	2	3	4	5	6	7	8	9
2013	5.067,76	23.306,17	38.203,07	46.336,78	50.346,93	51.729,36	53.227,74	53.621,64	53.634,61
2014	5.417,13	24.885,06	40.247,78	48.005,53	51.813,01	53.118,99	54.055,54	54.223,43	54.816,47
2015	5.655,37	24.073,24	38.990,56	47.505,09	50.545,90	52.004,93	53.008,00	53.672,75	54.304,62
2016	5.423,32	24.951,66	40.001,47	49.096,01	53.180,33	54.378,36	54.989,22	55.568,00	56.130,91
2017	6.071,79	28.323,80	44.555,86	53.026,78	57.733,74	58.451,31	58.919,41	59.353,56	59.786,65
2018	6.042,22	26.677,45	43.164,37	51.333,19	52.812,83	53.171,92	53.492,66	53.810,43	54.124,14
2019	5.314,01	24.778,95	40.532,81	42.084,36	42.538,77	42.866,13	42.967,24	43.168,29	43.369,37
2020	6.384,52	26.876,04	30.364,01	30.865,21	31.147,79	31.390,75	31.626,26	31.861,76	32.097,28
2021	5.701,45	6.704,26	7.087,92	7.213,92	7.287,15	7.349,27	7.410,92	7.472,568	7.534,45

Tabla 11. Triángulo de desarrollo acumulado con el método Random Forest (miles €). Elaboración propia.

El triángulo acumulado permite visualizar de manera clara cómo se distribuyen los pagos esperados a lo largo del tiempo para cada cohorte. En general, se observa que la mayoría de los pagos se concentran en los primeros años de desarrollo, aunque el modelo tiende a proyectar una cola larga en algunos casos.

A partir del triángulo completo, se extrajo el vector de pagos futuros agregados por año de calendario.

Año Dev.	2	3	4	5	6	7	8	9
Pagos futuros estimados	57.204,56	10.108,19	3.376,54	1.954,32	1.268,16	812,38	498,24	297,17

Tabla 12. Pagos futuros estimados por año de desarrollo (en €). Elaboración propia.

El importe total proyectado de pagos futuros utilizando el modelo de *Random Forest* asciende a 75.519,56 € (en miles de euros). Aplicando los factores de descuento correspondientes, se obtiene una provisión técnica descontada de 73.191,09 € (en miles de euros). Estos resultados constituyen una estimación clave para la planificación financiera y el análisis de suficiencia de reservas.

Si bien *Random Forest* ofrece una buena capacidad predictiva y un bajo error de ajuste en el conjunto de prueba (NRMSE ≈ 0.0056), presenta ciertas limitaciones en la proyección de pagos futuros. La falta de variabilidad en los años de desarrollo más avanzados indica que el modelo podría estar limitado por la escasez de datos representativos en esas regiones del triángulo. Se podría considerar ampliar la muestra de entrenamiento o aumentar el número de árboles para mejorar la generalización.

3.2.1.2.2. RESULTADOS CON MODELOS PENALIZADOS: LASSO

El modelo Lasso se empleó con el objetivo de realizar una selección automática de variables relevantes y evitar el sobreajuste, es especialmente útil en contextos donde se dispone de un número elevado de variables explicativas o donde algunas de ellas pueden ser redundantes. Su capacidad para imponer penalizaciones sobre los coeficientes permite simplificar el modelo, mejorando así su interpretabilidad sin perder capacidad predictiva.

Para ajustar el modelo, se utilizaron las variables explicativas transformadas mediante *model.matrix*, lo que permitió codificar adecuadamente las variables categóricas presentes en la base de datos. El modelo se entrenó utilizando validación cruzada con 5 particiones, obteniéndose el valor óptimo del parámetro de regularización lambda mediante minimización del error cuadrático medio.

Una vez entrenado el modelo con el valor óptimo de penalización (λ_min), se aplicó sobre las observaciones correspondientes a los años de desarrollo futuros.

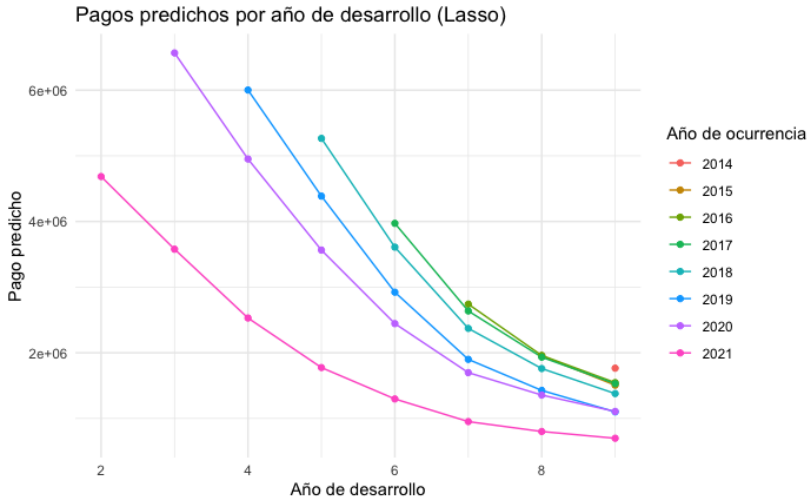

Ilustración 5. Pagos predichos por año de desarrollo con Lasso. Elaboración propia.

El gráfico muestra un patrón decreciente de pagos predichos por año de desarrollo, coherente con la evolución temporal esperada de los siniestros. Las cohortes más recientes (2020-2021) presentan importes significativos en los primeros años de desarrollo, con una reducción progresiva en los años siguientes. En cambio, para las cohortes más antiguas (2014-2016), solo se observan pagos residuales en los últimos años, reflejando que ya se han cubierto la mayoría de los costes asociados. En conjunto, el modelo Lasso genera una estructura triangular razonable y suavizada, aunque algo rígida en las extrapolaciones más alejadas, lo que es característico de este tipo de regularización.

A partir de las predicciones obtenidas, se completó el triángulo de desarrollo incluyendo los valores proyectados en las celdas faltantes para cada cohorte.

TRIÁNGULO DE DESARROLLO COMPLETO – LASSO

AÑO DE ORIGEN	AÑO DE DESARROLLO 1	2	3	4	5	6	7	8	9
2013	5.067,76	18.238,4	14.896,90	8.133,71	4.010,15	1.382,42	1.498,38	393,90	12.97
2014	5.417,13	19.467,94	15.362,71	7.757,76	3.807,48	1.305,98	936,55	167,89	1.765,34
2015	5.655,37	18.417,87	14.917,33	8.514,53	3.040,80	1.459,03	1.003,08	1.953,90	1.506,46
2016	5.423,32	19.528,33	15.049,81	9.094,54	4.084,32	1.198,03	2.741,04	1.959,21	1.545,86
2017	6.071,79	22.252,00	16.232,07	8.470,92	4.706,96	3.971,58	2.636,89	1.933,17	1.529,26
2018	6.042,22	20.635,23	16.486,92	8.168,82	5.265,58	3.609,18	2.371,70	1.759,74	1.378,66
2019	5.314,01	19.464,94	15.753,86	6.002,44	4.386,20	2.922,75	1.899,58	1.426,73	1.100,31
2020	6.384,52	20.491,51	6.566,99	4.951,14	3.563,92	2.444,53	1.697,81	1.356,03	1.106,70
2021	5.701,45	4.684,29	3.577,54	2.529,75	1.774,65	1.296,99	951,95	800,87	698,03

Tabla 13. Triángulo de desarrollo completo con el método Lasso (miles €). Elaboración propia.

TRIÁNGULO DE DESARROLLO ACUMULADO – LASSO

AÑO	AÑO DE DESARROLLO 1	2	3	4	5	6	7	8	9
2013	5.067,76	23.306,17	38.203,07	46.336,78	50.346,93	51.729,36	53.227,74	53.621,64	53.634,61
2014	5.417,13	24.885,06	40.247,77	48.005,53	51.813,01	53.118,99	54.055,54	54.223,43	55.988,77
2015	5.655,37	24.073,24	38.990,56	47.505,09	50.545,90	52.004,93	53.008,00	54.961,90	56.468,36
2016	5.423,32	24.951,66	40.001,47	49.096,01	53.180,33	54.378,36	57.119,40	59.078,61	60.624,46

2017	6.071,79	28.323,80	44.555,86	53.026,78	57.733,74	61.705,32	64.342,20	66.275,38	67,804,64
2018	6.042,22	26.677,45	43.164,37	51.333,19	56.598,76	60.207,94	62.579,64	64.339,38	65.718,04
2019	5.314,01	24.778,95	40.532,81	46.535,24	50.921,44	53.844,19	55.743,77	57.170,50	58.270,81
2020	6.384,52	26.876,04	33.443,03	38.394,17	41.958,08	44.402,61	46.100,42	47.456,45	48.563,15
2021	5.701,45	10.385,74	13.963,28	16.493,03	18.267,68	19.564,67	20.516,62	21.317,49	22.015,52

Tabla 14. Triángulo de desarrollo acumulado con el método Lasso (miles €). Elaboración propia

El triángulo acumulado generado por este modelo muestra una evolución coherente con la dinámica esperada de los pagos de siniestros, con una fuerte concentración en los primeros años de desarrollo y una disminución progresiva hacia los años posteriores. No obstante, se aprecia una mayor suavidad en los pagos proyectados respecto a modelos más complejos, lo que es característico de técnicas de regresión penalizada que tienden a estabilizar las predicciones y eliminar el ruido de variables no relevantes.

A partir del triángulo completo se extrajo el vector de pagos futuros agregados por año de calendario. Este vector se utilizó posteriormente para el cálculo de la provisión técnica descontada.

Año Dev.	2	3	4	5	6	7	8	9
Pagos futuros estimados	57.204,56	32.951,16	22.626,61	14.867,14	9.407,77	5.800,18	3.408,29	1.907,57

Tabla 15. Pagos futuros estimados por año de desarrollo (en €). Elaboración propia.

El importe total de pagos futuros estimado con el modelo *Lasso* asciende a 148.173,3 € (en miles de euros). Aplicando los factores de descuento adecuados —basados en la curva de rentas vencidas publicada por EIOPA a 5 de mayo de 2025— se obtiene una provisión técnica descontada de 140.436,5 € (en miles de euros).

En términos de precisión predictiva, el modelo Lasso presentó un buen resultado sobre el conjunto de prueba, con un NRMSE ≈ 0.0057, lo que refleja una capacidad razonable para ajustar la evolución acumulada de los pagos. Sin embargo, dada su naturaleza lineal y regularizadora, el modelo tiende a generar predicciones más conservadoras y menos adaptativas en las regiones del triángulo con mayor incertidumbre o escasez de datos, como los años de desarrollo más avanzados. A pesar de ello, el enfoque Lasso ofrece ventajas claras en términos de parquedad del modelo, control del sobreajuste y robustez frente a multicolinealidad, lo que lo convierte en una herramienta útil para tareas de predicción en entornos actuariales con estructuras de datos complejas.

3.2.1.2.3. RESULTADOS CON MODELOS LINEALES GENERALIZADOS

El modelo GLM Poisson con función de enlace logarítmica se utilizó por su comparabilidad con el caso estimado para GLM y datos agregados, donde se obtiene como caso particular CL. Aunque cabe notar que se pueden usar otras distribuciones como la Gamma o la Inversa Gaussiana. A diferencia de modelos no paramétricos, el GLM requiere especificar una forma funcional concreta y asume una relación log-lineal entre las variables explicativas y la media de la variable respuesta. Esta característica lo hace menos flexible, pero más interpretable y fácil de ajustar cuando se dispone de suficientes datos y se cumplen los supuestos del modelo.

Una vez entrenado el modelo sobre los pagos observados, se aplicó sobre el conjunto de datos con años de desarrollo futuros para obtener las predicciones.

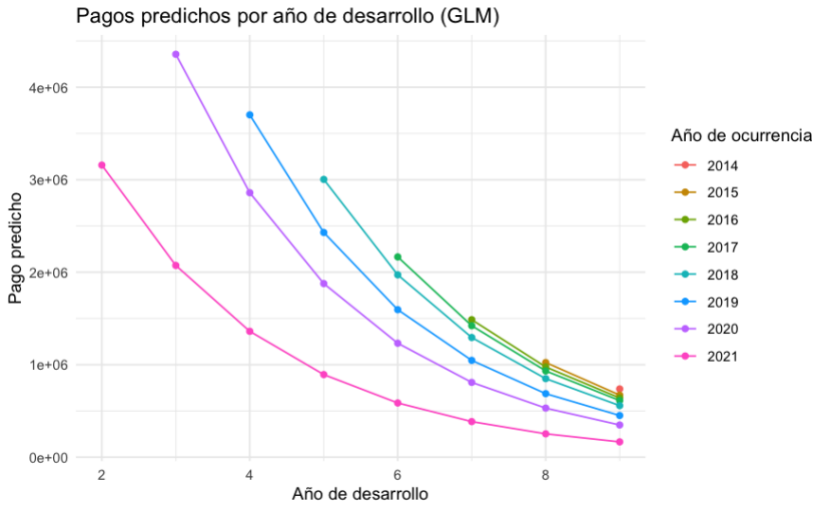

Ilustración 6. Pagos predichos por año de desarrollo con GLM. Elaboración propia.

El modelo GLM proyecta una evolución decreciente de los pagos conforme avanzan los años de desarrollo, concentrando los importes más elevados en los primeros años tras la ocurrencia del siniestro. Este patrón refleja adecuadamente el comportamiento típico del negocio asegurador, mostrando además un mayor volumen de pagos para los últimos años de ocurrencia más recientes, como 2020 y 2021. A diferencia de modelos más flexibles como *Random Forest*, el GLM ofrece una estructura más regular y predecible, aunque con cierta tendencia a subestimar los pagos en los desarrollos más lejanos, donde las relaciones no lineales son más relevantes.

Este comportamiento es coherente con la naturaleza del GLM, que al asumir una relación log-lineal entre las variables, no captura con la misma precisión los efectos de interacción o la heterogeneidad presente en los datos. No obstante, su estabilidad y capacidad explicativa lo convierten en una herramienta útil para estimaciones conservadoras y como referencia frente a modelos más complejos.

TRIÁNGULO DE DESARROLLO COMPLETO – GLM

AÑO DE ORIGEN \ AÑO DE DESARROLLO	1	2	3	4	5	6	7	8	9
2013	5.067,76	18.238,4	14.896,90	8.133,71	4.010,15	1.382,42	1.498,38	393,90	12.97
2014	5.417,13	19.467,94	15.362,71	7.757,76	3.807,48	1.305,98	936,55	167,89	738,05
2015	5.655,37	18.417,87	14.917,33	8.514,53	3.040,80	1.459,03	1.003,08	1.023,61	671,90
2016	5.423,32	19.528,33	15.049,81	9.094,54	4.084,32	1.198,03	1.486,64	975,85	640,56
2017	6.071,79	22.252,00	16.232,07	8.470,92	4.706,96	2.165,51	1.421,46	933,07	612,48
2018	6.042,22	20.635,23	16.486,92	8.168,82	3.003,48	1.971,52	1.294,13	849,48	557,61
2019	5.314,01	19.464,94	15.753,86	3.701,35	2.429,61	1.594,82	1.046,86	687,17	451,07
2020	6.384,52	20.491,51	4.356,40	2.859,59	1.877,07	1.232,13	808,79	530,90	348,49
2021	5.701,45	3.158,21	2.073,08	1.360,80	893,24	586,33	384,88	252,64	165,83

Tabla 16. Triángulo de desarrollo completo con el método GLM (miles €). Elaboración propia.

TRIÁNGULO DE DESARROLLO ACUMULADO – GLM

AÑO \ AÑO DE DESARROLLO	1	2	3	4	5	6	7	8	9
2013	5.067,76	23.306,17	38.203,07	46.336,78	50.346,93	51.729,36	53.227,74	53.621,64	53.634,61
2014	5.417,13	24.885,06	40.247,77	48.005,53	51.813,01	53.118,99	54.055,54	54.223,43	54.961,48
2015	5.655,37	24.073,24	38.990,56	47.505,09	50.545,90	52.004,93	53.008,00	54.031,60	54.703,51
2016	5.423,32	24.951,66	40.001,47	49.096,01	53.180,33	54.378,36	55.864,99	56.840,84	57.481,40

2017	6.071,79	28.323,80	44.555,86	53.026,78	57.733,74	59.899,24	61.320,71	62.253,77	62.866,25
2018	6.042,22	26.677,45	43.164,37	51.333,19	54.336,67	56.308,19	57.602,31	58.451,79	59.009,40
2019	5.314,01	24.778,95	40.532,81	44.234,15	46.663,76	48.258,58	49.305,44	49.992,61	50.443,68
2020	6.384,52	26.876,04	31.231,44	34.092,03	36.969,11	37.201,24	38.010,02	38.540,92	38.889,40
2021	5.701,45	8.859,65	10.932,74	12.293,53	13.186,77	13.773,11	14.157,99	14.410,62	14.576,46

Tabla 17. Triángulo de desarrollo acumulado con el método GLM (miles €). Elaboración propia

El triángulo acumulado resultante muestra que la mayor parte de los pagos estimados se concentran en los primeros años de desarrollo, como es habitual en el negocio asegurador. No obstante, en los últimos años de desarrollo, el GLM tiende a proyectar pagos más contenidos en comparación con *Random Forest*, lo que podría reflejar una cierta infraestimación debida a la falta de flexibilidad del modelo.

A partir del triángulo completo, se agregaron los pagos futuros por año calendario:

Año Dev.	2	3	4	5	6	7	8	9
Pagos futuros estimados	57.204,56	19.633,23	1.2403,02	7.700,44	4.634,19	2.639,90	1.366,84	601,12

Tabla 18. Pagos futuros estimados por año de desarrollo (en €). Elaboración propia.

El importe total proyectado de pagos futuros utilizando el modelo GLM asciende a 106.183,30 € (en miles de euros). Al aplicar los factores de descuento correspondientes (curva de rentas vencidas al 5 de mayo de 2025), se obtiene una provisión técnica descontada de 101.664,00 € (en miles de euros). Aunque inferior a la estimada por *Lasso*, esta cifra resulta razonable bajo la lógica conservadora del modelo.

En términos de precisión predictiva, el modelo GLM obtuvo un NRMSE ≈ 0.005769, ligeramente superior al de *Random Forest*. Esto indica una menor precisión en los valores individuales, si bien el GLM presenta mayor estabilidad en las estimaciones globales. Las posibles mejoras en este modelo pasarían por incorporar términos de interacción o explorar distribuciones alternativas (por ejemplo, una distribución *gamma* o *quasi-Poisson*) para capturar mejor la dispersión observada.

3.2.1.2.4. RESULTADOS CON K VECINOS MÁS CERCANOS

El modelo KNN se aplicó como alternativa no paramétrica, basada en la proximidad entre observaciones en un espacio multidimensional. Esta técnica no asume ninguna forma funcional específica para la relación entre las variables independientes y la variable respuesta, lo que le permite adaptarse a estructuras de datos complejas y potencialmente no lineales. Sin embargo, su rendimiento depende críticamente de la elección de los parámetros (especialmente el número de vecinos, k) y de la escala y representación de las variables explicativas.

Para aplicar KNN a la predicción de pagos, se seleccionaron como predictoras variables relevantes. Las variables categóricas fueron transformadas en variables *dummy* para poder operar en el espacio euclídeo requerido por el algoritmo.

Una vez entrenado el modelo sobre el conjunto de entrenamiento, se predijo el valor de los pagos acumulados para las combinaciones de años de origen y desarrollo futuros, reconstruyendo así el triángulo completo.

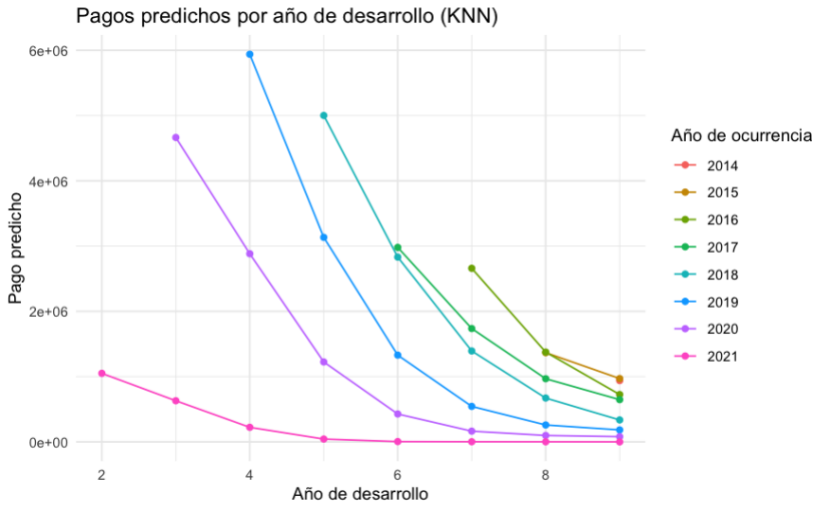

Ilustración 7. Pagos predichos por año de desarrollo (KNN). Elaboración propia.

El modelo KNN refleja una evolución de pagos algo menos regular que en modelos paramétricos, presentando ciertas oscilaciones entre años y desarrollos. No obstante, se observa una concentración notable de los pagos en los primeros años de desarrollo, en línea con la dinámica típica del negocio asegurador. En particular, los años de ocurrencia más recientes (2019, 2020 y 2021) muestran importes crecientes en los primeros años de desarrollo, aunque con una mayor dispersión en años posteriores.

A diferencia del GLM, el modelo KNN permite capturar relaciones complejas sin necesidad de especificar una estructura funcional, lo que resulta en mayor flexibilidad. Sin embargo, esta flexibilidad puede derivar en una menor estabilidad en las predicciones y una sensibilidad elevada a la calidad y escala de las variables de entrada, lo que justifica la variabilidad observada en el triángulo proyectado.

TRIÁNGULO DE DESARROLLO COMPLETO – KNN

AÑO DE ORIGEN \ AÑO DE DESARROLLO	1	2	3	4	5	6	7	8	9
2013	5.067,76	18.238,4	14.896,90	8.133,71	4.010,15	1.382,42	1.498,38	393,90	12.97
2014	5.417,13	19.467,94	15.362,71	7.757,76	3.807,48	1.305,98	936,55	167,89	939,41
2015	5.655,37	18.417,87	14.917,33	8.514,53	3.040,80	1.459,03	1.003,08	1.366,95	970,66
2016	5.423,32	19.528,33	15.049,81	9.094,54	4.084,32	1.198,03	2.659,73	1.375,71	725,30
2017	6.071,79	22.252,00	16.232,07	8.470,92	4.706,96	2.979,68	1.736,67	967,56	646,99
2018	6.042,22	20.635,23	16.486,92	8.168,82	5.001,31	2.831,48	1.392,93	671,55	336,04
2019	5.314,01	19.464,94	15.753,86	5.939,35	3.133,72	1.328,97	543,73	259,00	183,12
2020	6.384,52	20.491,51	4.664,02	2.882,44	1.224,48	427,16	164,05	98,25	80,12
2021	5.701,45	1.050,09	629,90	222,96	43,26	4,58	2,10	1,20	0,0000

Tabla 19. Triángulo de desarrollo completo con el método KNN (miles €). Elaboración propia.

TRIÁNGULO DE DESARROLLO ACUMULADO – KNN

AÑO DE \ AÑO DE DESARROLLO	1	2	3	4	5	6	7	8	9
2013	5.067,76	23.306,17	38.203,07	46.336,78	50.346,93	51.729,36	53.227,73	53.621,64	53.634,61
2014	5.417,13	24.885,06	40.247,78	48.005,53	51.813,01	53.118,99	54.055,54	54.223,43	55.162,84
2015	5.655,37	24.073,24	38.990,56	47.505,09	50.545,90	52.004,73	53.008,00	54.374,96	55.345,62
2016	5.423,32	24.951,66	40.001,47	49.096,01	53.180,33	54.378,36	57.038,09	58.413,80	53.139,11
2017	6.071,79	28.323,80	44.555,86	53.026,77	57.733,74	60.713,42	62.450,09	63.417,65	64.064,65
2018	6.042,22	26.677,45	43.164,37	51.333,19	56.334,49	59.165,97	60.558,90	61.230,45	61.566,48
2019	5.314,01	24.778,95	40.532,81	45.472,16	49.605,88	50.934,86	51.478,58	51.737,58	51.920,70

2020	6.384,52	26.876,04	31.540,10	34.422,50	35.646,97	36.074,13	36.238,19	36.336,43	36.416,55
2021	5.701,45	6.751,54	7.381,44	7.604,40	7.647,66	7.652,24	7.654,35	7.655,55	7.655,55

Tabla 20. Triángulo de desarrollo acumulado con el método KNN (miles €). Elaboración propia

El triángulo acumulado evidencia que la mayoría de los pagos se concentran en los primeros tres o cuatro años de desarrollo, aunque con una variabilidad ligeramente superior a la observada en GLM. Esta dispersión es esperable en modelos no paramétricos, y puede interpretarse como una consecuencia de la dependencia local del algoritmo: la predicción para cada punto depende directamente de la estructura y calidad de los datos cercanos en el espacio de covariables.

Al igual que en el caso anterior, se agregaron los pagos proyectados futuros por año de desarrollo:

Año Dev.	2	3	4	5	6	7	8	9
Pagos futuros estimados	57.204,56	24.600,53	13.560,60	5.862,21	2.332,69	763,67	283,47	81,32

Tabla 21. Pagos futuros estimados por año de desarrollo (en €). Elaboración propia.

El total proyectado de pagos futuros con KNN asciende a 104.689,10 € (en miles de euros), ligeramente superior a los obtenidos con GLM. Tras aplicar los factores de descuento (curva de rentas vencidas al 5 de mayo de 2025), se obtiene una provisión técnica descontada de 100.805,00 €.

En cuanto a la precisión predictiva, el modelo KNN obtuvo un error NRMSE de aproximadamente 0,0064, lo que sugiere un desempeño competitivo frente a modelos más estructurados como GLM. No obstante, su menor interpretabilidad y sensibilidad a la dimensionalidad hacen recomendable su uso combinado con métodos más explicativos en un entorno productivo.

En términos generales, el KNN ofrece una alternativa potente para capturar patrones complejos en los pagos de seguros, especialmente en contextos con alta heterogeneidad y cuando se dispone de un volumen suficiente de datos históricos.

3.3. CÁLCULO DE RESERVAS CON MÉTODOS TRADICIONALES

Una vez exploradas las técnicas de ML para la estimación de reservas, el siguiente paso ha consistido en el cálculo de reservas técnicas utilizando métodos tradicionales sobre el triángulo de siniestros reales (con pagos expresados en miles de euros), estructurado por años de ocurrencia (filas) y años de desarrollo (columnas).

3.3.1. MÉTODOS TRADICIONALES

Con los datos preparados y organizados en el formato de triángulo de pagos, se procede a aplicar dos métodos tradicionales para el cálculo de reservas: el método *Mack Chain-Ladder* y el modelo lineal generalizado con distribución *Poisson*. Estos métodos se basan en el análisis del triángulo de pagos acumulados y en la proyección de los costes futuros de los siniestros a partir de los patrones observados en los datos históricos.

Antes de proceder al desarrollo de los métodos deterministas y estocásticos, se presenta a continuación, en la Tabla 22, el triángulo completo de pagos incrementales, que servirá como base común para la estimación de las reservas. Este triángulo incluye tanto los pagos observados como los proyectados para los distintos años de origen y desarrollos. A

partir de él se derivarán las reservas futuras bajo las diferentes metodologías que se expondrán posteriormente.

TRIÁNGULO DE DESARROLLO COMPLETO – MÉTODOS TRADICIONALES

AÑO DE ORIGEN	AÑO DE DESARROLLO								
	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**
2013	5.067,76	18.238,4	14.896,90	8.133,71	4.010,15	1.382,42	1.498,38	393,90	12.97
2014	5.417,13	19.467,94	15.362,71	7.757,76	3.807,48	1.305,98	936,55	167,89	13,12
2015	5.655,37	18.417,87	14.917,33	8.514,53	3.040,80	1.459,03	1.003,08	277,57	12,89
2016	5.423,32	19.528,33	15.049,81	9.094,54	4.084,32	1.198,03	1.191,90	290,99	13,51
2017	6.071,79	22.252,00	16.232,07	8.470,92	4.706,96	1.498,95	1.298,30	316,97	14,72
2018	6.042,22	20.635,23	16.486,92	8.168,82	4.134,45	1.440,12	1.247,34	304,53	14,14
2019	5.314,01	19.464,94	15.753,86	8.289,69	3.932,23	1.369,68	1.186,33	289,63	13,45
2020	6.384,52	20.491,51	16.505,51	8.872,31	7.208,60	1.465,95	1.269,71	309,99	14,39
2021	5.701,45	19.914,53	15.731,85	8.456,44	4.011,33	1.397,23	1.210,20	295,46	13,72

Tabla 22. Triángulo de desarrollo completo con los métodos tradicionales (miles €). Elaboración propia.

3.3.1.1. *MACK CHAIN-LADDER*

El método *Mack Chain-Ladder* se basa en la extrapolación de los factores de desarrollo de los pagos (LDFs) para proyectar los costes futuros de los siniestros. Para su implementación, se utiliza la función *MackChainLadder()* del paquete *ChainLadder* (Gesmann et al., 2023), aplicando sobre el triángulo de pagos acumulados.

El modelo de Mack Chain-Ladder es un enfoque estocástico que extiende el método determinista CL, permitiendo la estimación de la variabilidad de las reservas. Este método parte del triángulo acumulado de pagos históricos y calcula los factores de desarrollo promedio entre columnas. Posteriormente, se proyectan los valores pendientes completando el triángulo hasta su madurez, generando el triángulo estimado completo. La diferencia entre cada celda del triángulo proyectado y su valor anterior (en la diagonal) permite obtener el triángulo desacumulado, es decir, los pagos futuros por año de ocurrencia y desarrollo.

A partir del triángulo desacumulado, se construyó el vector de pagos futuros, que resume los flujos de caja esperados para cada uno de los próximos años de desarrollo. La suma total de estos pagos futuros ascendió a aproximadamente 110.828,00 mil euros. Para reflejar el valor actual de estos flujos, se aplicó un descuento financiero utilizando la estructura de tipos de interés libre de riesgo ajustada por volatilidad publicada por EIOPA a 5 de mayo de 2025, bajo el supuesto de renta vencida. La provisión final estimada bajo este enfoque fue de 106.400,20 mil euros, reflejando el valor presente de las obligaciones futuras de la aseguradora.

3.3.1.2. MODELO LINEAL GENERALIZADO CON DISTRIBUCIÓN *POISSON*

El segundo enfoque aplicado fue un modelo lineal generalizado (GLM) que permite una formulación más flexible de los pagos de siniestros. En este caso, se asumió una distribución *Poisson* sobre dispersa, apropiada para datos pagos agregados, y se empleó una función de enlace logarítmica. El modelo se ajustó directamente sobre el triángulo de pagos acumulados y permitió estimar el triángulo completo proyectado.

Al igual que con *Mack*, se desacumuló el triángulo estimado para obtener el flujo de pagos pendientes por cada año de desarrollo. Esto permitió construir su correspondiente vector

de pagos futuros, cuya suma total fue de aproximadamente 110.828,00 mil euros. Posteriormente, se aplicó el mismo proceso de descuento utilizando la curva de tipos de interés libre de riesgo, obteniéndose así una provisión actualizada bajo el enfoque GLM igual a 106.400,20 mil euros.

3.4. ANÁLISIS COMPARATIVO DE TÉCNICAS

El análisis comparativo de los métodos aplicados, incluyendo enfoques tradicionales y de aprendizaje automático, pone de manifiesto diferencias relevantes tanto en la estimación de los pagos futuros como en el importe de las provisiones calculadas bajo el supuesto de renta vencida.

Los métodos tradicionales como *Mack* CL y el GLM Tradicional (*Poisson*) ofrecen resultados idénticos, tanto en términos de pagos futuros (aproximadamente 110,8 millones de euros) como de provisión bajo renta vencida (en torno a 106,4 millones). Esta similitud se justifica por la estructura común de ambos modelos, basada en la descomposición aditiva o multiplicativa del desarrollo temporal de los siniestros. Mientras *Mack* incorpora una componente estocástica que permite estimar la variabilidad de la reserva, el GLM clásico se mantiene en un marco puramente estadístico bajo la suposición de una distribución *Poisson*, pero sin introducir incertidumbre explícita. En ambos casos, los resultados muestran consistencia, robustez y estabilidad en contextos donde los patrones históricos son regulares.

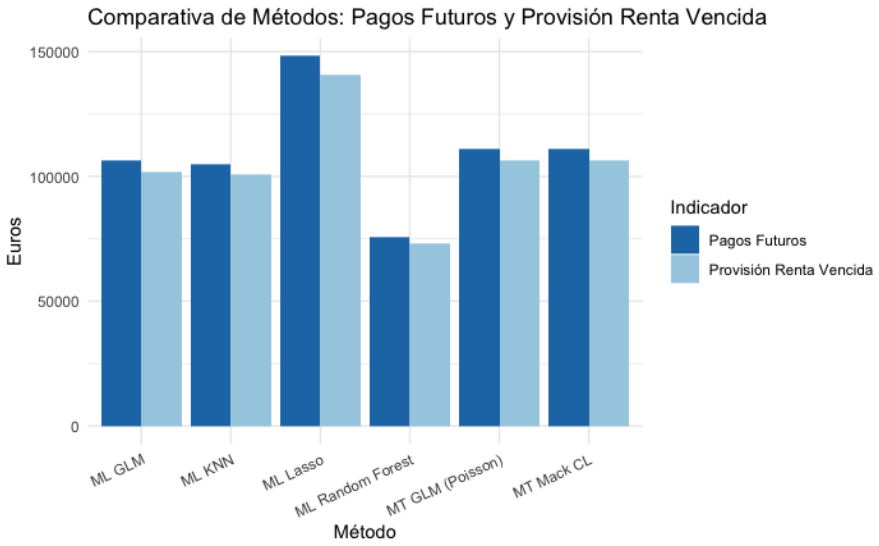

Ilustración 8. Pagos futuros y provisión Renta Vencida de los diferentes métodos

En contraposición, con los métodos de aprendizaje automático se obtienen estimaciones más heterogéneas. El modelo de *Random Forest*, por ejemplo, estima unos pagos futuros en torno a los 76,6 millones y una provisión de 74,1 millones, situándose muy por debajo de los métodos tradicionales. Este comportamiento puede deberse a la capacidad del modelo para identificar patrones específicos y no lineales en los datos, así como a su posible tendencia a infraestimar los pagos futuros si no se capturan adecuadamente siniestros de cola larga o con retrasos en el reporte.

Por otro lado, el modelo *Lasso* se encuentra en el extremo opuesto, ofreciendo una estimación muy superior al resto de métodos: los pagos futuros ascienden a 148,1 millones de euros y la provisión a 140,4 millones. Esta sobreestimación puede deberse a

una sensibilidad mayor ante covariables relevantes o a una menor penalización de desarrollos extremos, lo cual puede ser útil en contextos con riesgos catastróficos o alta variabilidad, pero también puede generar un sesgo si el modelo está sobre ajustado.

El GLM ML, que introduce elementos propios del ML (como optimización automática o selección de variables con penalización), proporciona una estimación intermedia: 106,2 millones de euros en pagos futuros y 101,6 millones en provisión. Este modelo actúa como un puente entre los métodos tradicionales y los de ML, capturando mejor la estructura de los datos sin incurrir en las desviaciones extremas de otros algoritmos.

Finalmente, el modelo KNN ofrece una estimación ligeramente inferior a la de *Mack*: 104,7 millones de euros en pagos y 100,8 millones en provisión. Su comportamiento se caracteriza por seguir de cerca los desarrollos observados en años con características similares, lo que puede resultar útil en entornos estables pero problemático ante eventos inesperados o cambios de tendencia.

En conjunto, estos resultados evidencian la importancia de seleccionar adecuadamente la metodología según el contexto: mientras los métodos tradicionales proporcionan resultados más estables y fácilmente interpretables, los enfoques de aprendizaje automático pueden detectar dinámicas complejas, pero requieren una validación cuidadosa para evitar sobreajustes o subestimaciones.

4. CONCLUSIONES

En este trabajo se realiza la aplicación y comparativa de diversas técnicas para el cálculo de reservas en seguros, tanto de métodos tradicionales, como de técnicas modernas basadas en ML. A lo largo del análisis, se ha podido observar cómo cada enfoque presenta fortalezas y limitaciones específicas, lo que refuerza la idea de que no existe una solución única o universal para la cuantificación del riesgo en el ámbito actuarial.

Los métodos tradicionales siguen siendo un pilar fundamental en la práctica actuarial debido a su transparencia, estabilidad y facilidad de interpretación, especialmente en entornos regulatorios que demandan justificación y trazabilidad. Sin embargo, estas metodologías tienden a asumir ciertas condiciones, como la independencia condicional y patrones lineales en el desarrollo de siniestros, que no siempre reflejan la realidad compleja y cambiante que enfrentan las compañías de seguros.

Por otro lado, las técnicas de ML han demostrado ser herramientas que permiten capturar patrones más complejos y relaciones no lineales en los datos, lo que puede traducirse en una mejor adaptación a situaciones con alta volatilidad, cambios estructurales en el mercado o nuevos riesgos emergentes. No obstante, la implementación de estos modelos requiere un manejo cuidadoso para evitar problemas de sobreajuste, falta de interpretabilidad y dependencia excesiva de la calidad y cantidad de datos disponibles. La combinación de modelos estadísticos y técnicas de ML representa una vía prometedora para mejorar la precisión y utilidad práctica de las provisiones, siempre bajo el prisma de una validación rigurosa y un juicio experto.

En el contexto actual, caracterizado por una creciente incertidumbre económica y social, influida por fenómenos como la pandemia global, el cambio climático, la digitalización acelerada y la evolución de los perfiles de riesgo de los asegurados, es esencial que los actuarios cuenten con herramientas flexibles y avanzadas para anticipar y gestionar estos desafíos. La capacidad de integrar diferentes enfoques metodológicos, adaptándose a las particularidades de la cartera y del entorno, se convierte en una ventaja competitiva y en un requisito indispensable para garantizar la sostenibilidad financiera y la solvencia de las instituciones.

Además, este estudio pone de manifiesto la importancia de la formación continua en nuevas tecnologías y métodos cuantitativos dentro de la profesión actuarial. La rápida evolución de las técnicas analíticas y la disponibilidad creciente de datos requieren que los profesionales se mantengan actualizados y desarrollen habilidades en programación, análisis de datos y modelización avanzada. Elfuturi de la cuantificación del riesgo en seguros estará basado en una combinación que aproveche la solidez y claridad de los métodos tradicionales junto con la innovación y flexibilidad que ofrecen las técnicas de ML. Esta unión permitirá enfrentar de manera más efectiva los desafíos que vienen, contribuyendo a mejorar la protección de los asegurados y a fortalecer la estabilidad del sector asegurador en general.

5. BIBLIOGRAFÍA Y WEBGRAFÍA

Ahlgren, M. (2018). *Claims reserving using Gradient Boosting and Generalized Linear Models* [Master's thesis, KTH Royal Institute of Technology]. https://www.diva-portal.org/smash/get/diva2:1230916/FULLTEXT01.pdf

Avanzi, B., Taylor, G., Wangb, M., Wongb, B. (2021). SynthETIC: an individual insurance claim simulator with feature control. Insurance: Mathematics and Economics 100, 296-308. https://doi.org/10.1016/j.insmatheco.2021.06.004

Breiman, L. (2001). Random Forests. Machine Learning, 45(1), 5–32.

Cover, T., Hart, P. (1967). Nearest neighbor pattern classification. IEEE Transactions on Information Theory, 13(1), 21–27.

Datacamp (2024, 29 de abril). Explicación de las funciones de pérdida en el machine learning. Datacamp. https://www.datacamp.com/es/tutorial/loss-function-in-machine-learning

De Felice, M., Moriconi, F. (2019). Claim watching and individual claims reserving using classification and regression trees. Insurance: Mathematics and Economics, 89, 28–41. https://doi.org/10.1016/j.insmatheco.2019.01.002

Duval, F., Pigeon, M. (2019). *Gradient Boosting-Based Model for Individual Loss Reserving*. https://arxiv.org/abs/1911.02364

EIOPA (2025). Risk-Free Interest Rate Term Structures with Volatility Adjustment – 5 May 2025. European Insurance and Occupational Pensions Authority. https://www.eiopa.europa.eu

Gesmann, M., Murphy, D., Zhang, Y., Carrato, A., Wuthrich, M.V., Concina, F., Dal Moro, E. (2022). ChainLadder: statistical methods and models for claims reserving in general insurance. R package version 0.2.15. https://CRAN.R-project.org/package=ChainLadder

Hastie, T., Tibshirani, R., y Friedman, J. (2009). The Elements of Statistical Learning: Data Mining, Inference, and Prediction. 2nd Edition. Springer.

Lozano, M., González, M. (2010). Métodos estocásticos de estimación de las provisiones técnicas en el marco de Solvencia II. Fundación MAPFRE.

Lundberg, S. M., Lee, S.-I. (2017). A Unified Approach to Interpreting Model Predictions. In Advances in Neural Information Processing Systems (NeurIPS 2017), 30, 4765–4774. https://proceedings.neurips.cc/paper_files/paper/2017/file/8a20a8621978632d76c43dfd28b67767-Paper.pdf

Mack, T. (1993). Measuring the Variability of Chain Ladder Reserve Estimates, Spring Vol I, Casualty Actuarial Society E-Forum.

Nelder, J. A., Wedderburn, R. W. M. (1972). Generalized Linear Models. Journal of the Royal Statistical Society: Series A (General), 135(3), 370–384.

R Development Core Team (2025). R: The R project for statistical computing. Vienna (Austria). http://www.R-project.org/.

Tibshirani, R. (1996). Regression shrinkage and selection via the Lasso. Journal of the Royal Statistical Society: Series B (Methodological), 58(1), 267–288.

Wüthrich, M. V., Merz, M. (2008). Stochastic Claims Reserving Methods in Insurance. Wiley.

6. ANEXOS

6.1. ANEXOS (CÓDIGO DE R)

````
---
title: "TFM"
author: "Patricia López Ozcoz"
date: "`r Sys.Date()`"
output: html_document
---


# SIMULACIÓN DE LA BASE DE DATOS DE SINIESTROS

Librerias


```{r, echo=FALSE, include=FALSE}
# Establecimiento de directorio de trabajo
setwd("/Users/patrilopez/Desktop/UNI/Máster/2o año/2º Semestre/TFM/Datos R")
getwd()
## Paquetes a usar
install.packages("SynthETIC")
install.packages("plyr")
install.packages("locfit")
install.packages("dplyr")
install.packages("actuar")
install.packages("tidyr")
install.packages("reshape2")
install.packages("ChainLadder", dependencies = TRUE)
install.packages('Rcpp')
install.packages("dplyr")
install.packages("ggplot2")
install.packages("lattice")
install.packages("caTools")
install.packages('openxlsx')
```

```{r}
library(SynthETIC)
library(dplyr)
library(openxlsx)
library(tidyverse)
library(skimr)
library(GGally)
library(corrplot)
library(DataExplorer)
library(ggplot2)
library(tidyr)
library(randomForest)
library(caret)
library(glmnet)
library(gbm)
library(nnet)
library(class)
library(reshape2)
library(openxlsx)
library(ChainLadder)
library(writexl)
```

```


Base de datos

```{r}
ref_claim = 1 # divisa = 1
time_unit = 1/12 # Consideramos el desarrollo de siniestros mensuales
set_parameters(ref_claim = ref_claim, time_unit = time_unit)
years = 10 # número de años de ocurrencia
I = years/time_unit # número de periodos
source("./functions simulation.R")
PathData = "./Data/"
PathResults = "./Results/"
PathPlot = "./Plots/"
claims_list = data.generation(seed = 123, future_info = FALSE)
siniestros = claims_list[[1]]
pagos = claims  list[[2]]
```


Tratamiento adicional de la base de datos simulada
```{r}
siniestros$AccYearMonth = as.integer(x = format(x = siniestros$AccDate,'%m'))
siniestros$RepYearMonth = as.integer(x = format(x = siniestros$RepDate,'%m'))
siniestros$SetDelDays = as.numeric(x = siniestros$SetDelMonths * 30)
siniestros$RepDelMonths = as.numeric(x = siniestros$RepDelDays / 30)
siniestros$AccYear = as.integer(x = format(x = siniestros$AccDate,'%Y'))
siniestros$RepYear = as.integer(x = format(x = siniestros$RepDate,'%Y'))

write.csv(x = siniestros, file = paste(PathData, "siniestros.csv", sep = ""))
write.csv(x = pagos, file = paste(PathData, "pagos.csv", sep = ""))

if (!dir.exists(PathData)) {
  dir.create(PathData)
}
write.csv(x = siniestros, file = paste(PathData, "siniestros.csv", sep = ""))
write.csv(x = pagos, file = paste(PathData, "pagos.csv", sep = ""))
```


Datos ya generados y guardados
```{r}
siniestros <- read.csv("~/Desktop/UNI/Master/2o año/2º Semestre/TFM/Datos R/Data/siniestros 3.csv")
pagos <- read.csv("~/Desktop/UNI/Master/2o año/2º Semestre/TFM/Datos R/Data/pagos 3.csv")
```

# ANÁLISIS DESCRIPTIVO

```{r}
siniestros  limpio <- siniestros %>%
  filter(Status %in% c("Closed", "RBNS"))

table(siniestros_limpio$Status)
summary(siniestros_limpio)
```

*ANÁLISIS DESCRIPTIVO GENERAL*

```{r}
skim(siniestros_limpio)
introduce(siniestros_limpio)
plot_intro(siniestros_limpio)
```

BBDD SINIESTROS
```{r}
# Estructura y primeras filas
str(siniestros)
head(siniestros)

# Estadísticos de variables numéricas
summary(siniestros)

# Tablas de frecuencias para variables categóricas
table(siniestros$Type)
table(siniestros$Status)
table(siniestros$AccMonth)
table(siniestros$AccWeekday)

# Histograma de edades
````

```
hist(siniestros$Age, main =
"Distribución de Edad", xlab =
"Edad", col = "skyblue")
```

BBDD PAGOS
```{r}
# Estructura y primeras filas
str(pagos)
head(pagos)

# Estadísticos de la variable
Paid
summary(pagos$Paid)

# Histograma de pagos (puedes
truncar valores extremos si hay
outliers)
hist(pagos$Paid, breaks = 100,
main = "Distribución de Pagos",
xlab = "Paid", col =
"lightgreen", xlim = c(0,
quantile(pagos$Paid, 0.99,
na.rm = TRUE)))

# Proporción de pagos negativos
(antes de truncar)
mean(pagos$Paid < 0, na.rm =
TRUE)
```

PREPARACIÓN DE DATOS Y CONSTRUCCIÓN PANEL INCREMENTAL

```{r}
siniestros$Id <-
as.numeric(siniestros$Id)
pagos$Id <-
as.numeric(pagos$Id)

# Año de ocurrencia y de pago
siniestros_p4 <- siniestros %>%
  mutate(AccYear =
year(AccDate))

# Año de Pago en bbdd pagos
pagos_p5 <- pagos %>%
  mutate(PayYear =
floor((EventMonth - 1)/12) +
2012) %>%
  left_join(siniestros_p4 %>%
select(Id, AccYear), by = "Id")
%>%
  mutate(DevYear = PayYear -
AccYear + 1) %>%
  filter(DevYear >= 1 &
DevYear <= 10, AccYear >
2012)

# Pagos incrementales por
siniestro y año de desarrollo
pagos_agg5 <- pagos_p5 %>%
  group_by(Id, AccYear,
DevYear) %>%
  summarise(CumPaid =
sum(Paid, na.rm = TRUE))
%>%
  ungroup() %>%
  mutate(CumPaid =
pmax(CumPaid, 0))

# Base panel: Solo Ids cuyo
AccYear > 2012
panel5 <- expand.grid(
  Id = unique(siniestros_p4$Id),
  DevYear = 1:9
) %>%
  left_join(
    siniestros_p4 %>%
      filter(AccYear > 2012)
%>%
      select(Id, AccYear, Type,
Age, AccMonth, AccWeekday,
RepDelDays, Status),
    by = "Id"
  ) %>%
  arrange(Id, DevYear)

# Unir pagos acumulados (panel
pagos_agg5)
panel5 <- panel5 %>%
  left_join(pagos_agg5, by =
c("Id", "AccYear", "DevYear"))
%>%
  arrange(Id, DevYear)

panel5 <- panel5 %>%
  complete(Id, DevYear = 1:9,
fill = list(CumPaid = 0)) %>%
  mutate(CumPaid =
ifelse(is.na(CumPaid), 0,
CumPaid)) %>%
  mutate(CumPaid =
pmax(CumPaid, 0))

# Calcular pagos acumulados
panel5 <- panel5 %>%
  arrange(Id, DevYear) %>%
  group_by(Id) %>%
  mutate(CumPaidToDate =
cumsum(CumPaid)) %>%
  ungroup()

# Verificar duplicados
duplicados <- panel5 %>%
  count(Id, DevYear) %>%
  filter(n > 1)
if(nrow(duplicados) > 0)
print(duplicados)

# Filtrar solo estados Closed y
RBNS (por si acaso)
panel5 <- panel5 %>%
  filter(Status %in% c("Closed",
"RBNS"))

# Tablas y resumen
table(panel5$Status)
table(panel5$AccYear)
summary(panel5)

# Transformar a factores
panel5$Type <-
as.factor(panel5$Type)
panel5$Status <-
as.factor(panel5$Status)
panel5$AccWeekday <-
as.factor(panel5$AccWeekday)
panel5$Paid <-
panel5$CumPaid
```

ANÁLISIS DE LA VARIABLE OBJETIVO

```{r}
# Estadísticos descriptivos de
Paid
summary(panel5$Paid)
```

```{r}
library(dplyr)

# Tabla por año de origen
columnas_Paid <- panel5 %>%
  group_by(AccYear) %>%
  summarise(
    `Número de registros` = n(),
    `Número de siniestros únicos`
= n_distinct(Id),
    `Paid total (miles €)` =
round(sum(Paid, na.rm =
TRUE) / 1000, 2),
    `Promedio Paid (miles €)` =
round(mean(Paid, na.rm =
TRUE) / 1000, 2),
    `Mínimo Paid (miles €)` =
round(min(Paid, na.rm =
TRUE) / 1000, 2),
    `Máximo Paid (miles €)` =
round(max(Paid, na.rm =
TRUE) / 1000, 2)
  ) %>%
  mutate(`Año de origen` =
as.character(AccYear)) %>%  #
Convertimos a character
  select(`Año de origen`,
everything(), -AccYear)

# Fila total
fila_Paid <- panel5 %>%
  summarise(
    `Año de origen` = "Total",
    `Número de registros` = n(),
    `Número de siniestros únicos`
= n_distinct(Id),
    `Paid total (miles €)` =
round(sum(Paid, na.rm =
TRUE) / 1000, 2),
    `Promedio Paid (miles €)` =
round(mean(Paid, na.rm =
TRUE) / 1000, 2),
    `Mínimo Paid (miles €)` =
round(min(Paid, na.rm =
TRUE) / 1000, 2),
```

```
   `Máximo Paid (miles €)` =
round(max(Paid, na.rm =
TRUE) / 1000, 2)
  )

# Combinamos
tabla_Paid <-
bind_rows(columnas_Paid,
fila_Paid)

# Imprimir
print(tabla_Paid)


library(ggplot2)
tabla_Paid2 <- panel5 %>%
  group_by(AccYear, DevYear)
%>%
  summarise(
    TotalPaid = sum(Paid, na.rm
= TRUE),
    N_IDs = n_distinct(Id)
  ) %>%
  ungroup()

media_dev <- panel5 %>%
  group_by(DevYear) %>%
  summarise(MediaPaid =
mean(Paid, na.rm = TRUE))

ggplot(media_dev, aes(x =
DevYear, y = MediaPaid)) +
  geom_line() +
  geom_point() +
  labs(title = "Media de Paid por
Año de Desarrollo",
       x = "DevYear", y = "Media
Paid") +
  theme_minimal()


```


```{r}
# Proporción de ceros
mean(panel5$Paid == 0, na.rm
= TRUE)

# Filtrar valores mayores a 0
Paid_sin_ceros <-
panel5$Paid[panel5$Paid > 0]

# Histograma de Paid (sin ceros)
hist(Paid_sin_ceros, breaks =
200, main = "Distribución de
Paid (sin ceros)",
     xlab = "Paid", col =
"orange")

# Boxplot para detectar outliers
(sin ceros)
boxplot(Paid_sin_ceros, main =
"Boxplot de Paid (sin ceros)",
        horizontal = TRUE, col =
"gold")
```

```
# Estadísticos adicionales (sin
ceros)
sd(Paid_sin_ceros, na.rm =
TRUE)
quantile(Paid_sin_ceros, probs
= c(0.01, 0.25, 0.5, 0.75, 0.99),
na.rm = TRUE)

```


```{r}
library(dplyr)
library(ggplot2)
library(scales)

# Tabla resumen: suma de Paid
por AccYear y DevYear
tabla_Paid <- panel5 %>%
  group_by(AccYear, DevYear)
%>%
  summarise(
    TotalPaid = sum(Paid, na.rm
= TRUE),
    N_IDs = n_distinct(Id)
  ) %>%
  ungroup()

panel5 %>%
  group_by(AccYear) %>%
  summarise(
    TotalPaid = sum(Paid, na.rm
= TRUE),
    N_IDs = n_distinct(Id)
  ) %>%
  ungroup()

print(tabla_Paid)

```


```{r}
# Relación de Paid con años de
desarrollo y de ocurrencia
boxplot(Paid ~ DevYear, data =
panel5, main = "Paid por Año
de Desarrollo", col =
"lightblue")
boxplot(Paid ~ AccYear, data =
panel5, main = "Paid por Año
de Ocurrencia", col =
"lightpink")

```
```{r}
# 1. Filtrar los casos con Paid >
0
df_no_ceros <-
panel5[panel5$Paid > 0 &
!is.na(panel5$Age), ]

# 2. Crear grupos de edad
(ajusta los cortes si es necesario)
df_no_ceros$EdadGrupo <-
cut(df_no_ceros$Age,
                    breaks =
c(18, 30, 40, 50, 60, 70, 80, Inf),
                    right =
FALSE,
                    labels =
c("18-29", "30-39", "40-49",
"50-59", "60-69", "70-79",
"80+"))

# 3. Estadísticos descriptivos
por grupo de edad
library(dplyr)

resumen <- df_no_ceros %>%
  group_by(EdadGrupo) %>%
  summarise(
    n = n(),
    media = mean(Paid, na.rm =
TRUE),
    mediana = median(Paid,
na.rm = TRUE),
    sd = sd(Paid, na.rm = TRUE),
    p25 = quantile(Paid, 0.25,
na.rm = TRUE),
    p75 = quantile(Paid, 0.75,
na.rm = TRUE)
  )

print(resumen)

# Calcular la media por grupo
de edad
media_por_grupo <-
df_no_ceros %>%
  group_by(EdadGrupo) %>%
  summarise(media_Paid =
mean(Paid, na.rm = TRUE))

# Graficar
ggplot(media_por_grupo, aes(x
= EdadGrupo, y = media_Paid))
+
  geom_col(fill = "steelblue") +
  labs(title = "Media de Paid por
grupo de edad (sin ceros)",
       x = "Grupo de edad",
       y = "Media de Paid") +
  theme_minimal()
# Calcular la suma total por
grupo de edad
suma_por_grupo <- df_no_ceros
%>%
  group_by(EdadGrupo) %>%
  summarise(suma_Paid =
sum(Paid, na.rm = TRUE))

# Graficar
ggplot(resumen, aes(x =
EdadGrupo, y = media)) +
  geom_col(fill = "steelblue") +
  labs(title = "Media de Paid por
grupo de edad ",
       x = "Grupo de edad",
       y = "Suma total de Paid") +
  theme_minimal()
```

```
# 5. Scatterplot con suavizado LOESS
plot(df_no_ceros$Age, df_no_ceros$Paid,
     main = "Relación entre Edad y Paid (sin ceros)",
     xlab = "Edad", ylab = "Paid", pch = 20, col = rgb(0, 0, 0, 0.3))
lines(lowess(df_no_ceros$Age, df_no_ceros$Paid), col = "red", lwd = 2)

# 6. Correlaciones
cor(df_no_ceros$Paid, df_no_ceros$Age, use = "complete.obs")                    # Pearson
cor(df_no_ceros$Paid, df_no_ceros$Age, method = "spearman", use = "complete.obs") # Spearman

```

```{r}
# Comparación de medias por grupo para variables categóricas
aggregate(Paid ~ Type, data = panel5, mean)
aggregate(Paid ~ Status, data = panel5, mean)
aggregate(Paid ~ Status, data = panel5, sum)

# Boxplot para visualizar la relación con variables categóricas
boxplot(Paid ~ Type, data = panel5, main = "Paid por Tipo de Siniestro")
boxplot(Paid ~ Status, data = panel5, main = "Paid por Estado")

```
```{r}
table(panel5$Status)
```


ENTRENAMIENTO Y DE PRUEBA

```{r}
set.seed(123)
tamano_total5 <- nrow(panel5)
tamano_muestra5 <- round(tamano_total5 * 0.15) # 15%
indices_muestra5 <- sample(1:tamano_total5, size = tamano_muestra5)
train_sample5 <- panel5[indices_muestra5, ]
test  sample5 <- panel5[-indices_muestra5, ] # 85%

# Solo años de ocurrencia históricos para entrenamiento
train_sample5 <- train_sample5 %>% filter(AccYear <= 2021)
```
```{r}
train_sample5 %>%
  count(AccYear, DevYear, Status) %>%
  arrange(AccYear, DevYear)
# Miramos que no haya NA en variables
sum(is.na(train_sample5))
```

RANDOM FOREST

```{r}
train_sample5_RF<- train_sample5
test_sample5_RF<-test_sample5

rf_model5 <- randomForest(
  Paid ~ AccYear + DevYear + Type + Age + AccMonth + AccWeekday + RepDelDays + Status,
  data = train_sample5_RF,
  ntree = 30,
  importance = TRUE)

importance(rf_model5)
```

PREDICCIÓN DEL ERROR (TEST)

PREDICCIÓN

```{r}
# Predice pagos incrementales para el test
test_sample5_RF$Paid_pred <- predict(rf_model5, newdata = test_sample5_RF)

```


Cálculo del RMSE y NRMSE

```{r}
# Calcula el MSE
MSE_RF <- mean((test_sample5_RF$Paid - test_sample5_RF$Paid_pred)^2, na.rm = TRUE)

# RMSE
RMSE_RF <- sqrt(MSE_RF)

# NRMSE normalizado por el rango
max_ult <- max(test_sample5_RF$Paid, na.rm = TRUE)
min_ult <- min(test_sample5_RF$Paid, na.rm = TRUE)
NRMSE_RF <- RMSE_RF / (max_ult - min_ult)

# Mostrar resultados
cat("RMSE:", RMSE_RF, "\n")
cat("NRMSE:", NRMSE_RF, "\n")
```

BBDD Pagos Futuros
```{r}
# Para cada AccYear, identificar el máximo DevYear con datos reales
rf_max_dev_by_accyear5 <- panel5 %>%
  filter(AccYear <= 2021) %>%
  group_by(AccYear) %>%
  summarise(max_dev = max(DevYear[Paid > 0], na.rm = TRUE))
```

```{r}
# Unión del máximo DevYear al panel
rf_panel_futuro5 <- panel5 %>%
  filter(AccYear <= 2021) %>%

left_join(rf_max_dev_by_accyear5, by = "AccYear") %>%
  filter(DevYear > max_dev)
```

```{r}
table(rf_panel_futuro5$AccYear)
```

Predicción de pagos incrementales futuros

```{r}
rf_panel_futuro5$Paid_pred <- predict(rf_model5, newdata = rf_panel_futuro5)
rf_panel_futuro5$Paid_pred <- pmax(rf_panel_futuro5$Paid_pred, 0)
```

Construcción de triángulos

Triángulo real

```{r}
# Triángulo real (pagos históricos)
triangle_real5 <- panel5 %>%
  filter(AccYear <= 2021) %>%
  group_by(AccYear, DevYear) %>%
```

```
  summarise(Paid = sum(Paid,
na.rm = TRUE), .groups =
"drop")

triangle_real_wide5 <-
triangle_real5 %>%
  pivot_wider(
    names_from = DevYear,
    values_from = Paid,
    names_sort = TRUE
  )

print(triangle_real_wide5/1000)
```

Triángulo futuro

```{r}
#Triángulo futuro (pagos
predichos)
rf_triangle_pred_futuro5 <-
rf_panel_futuro5 %>%
  group_by(AccYear, DevYear)
%>%
  summarise(Paid =
sum(Paid_pred, na.rm = TRUE),
.groups = "drop") %>%
  tidyr::spread(key = DevYear,
value = Paid)

print(rf_triangle_pred_futuro5)
```

```{r}
ggplot(rf_triangle_pred_futuro5
%>% gather(DevYear, Paid, -
AccYear),
       aes(x =
as.numeric(DevYear), y = Paid,
group = AccYear, color =
as.factor(AccYear))) +
  geom_line() + geom_point() +
  labs(title = "Pagos predichos
por año de desarrollo RF", x =
"Año de desarrollo", y = "Pago
predicho")

rf_triangle_pred_futuro5_long
<- rf_triangle_pred_futuro5
%>%
  pivot_longer(
    cols = -AccYear,
    names_to = "DevYear",
    values_to = "Paid"
  ) %>%
  mutate(DevYear =
as.numeric(DevYear))

# Gráfico de líneas por año de
ocurrencia
ggplot(rf_triangle_pred_futuro5
_long, aes(x = DevYear, y =
Paid, color =
as.factor(AccYear))) +
  geom  line() +
  geom_point() +
  labs(
    title = "Pagos predichos por
año de desarrollo (Random
Forest)",
    x = "Año de desarrollo",
    y = "Pago predicho",
    color = "Año de ocurrencia"
  ) +
  theme_minimal()

```

*Triángulo combinado (real +
predicho)*
```{r}
# Triángulo futuro (pagos
predichos, ya truncados a cero)
rf_triangle_futuro5 <-
rf_panel_futuro5 %>%
  group_by(AccYear, DevYear)
%>%
  summarise(Paid =
sum(Paid_pred, na.rm = TRUE),
.groups = "drop")

# Combina ambos: si hay
predicción, úsala; si no, usa el
valor real
rf_triangle_completo5 <-
bind_rows(triangle_real5,
rf_triangle_futuro5) %>%
  group_by(AccYear, DevYear)
%>%
  summarise(Paid = sum(Paid,
na.rm = TRUE), .groups =
"drop") %>%
  tidyr::spread(key = DevYear,
value = Paid)

rf_triangle_completo5<-
rf_triangle_completo5/1000
print(rf_triangle_completo5)
#Desacumulado
```


Triangulo acumulado
```{r}
rf_triangle_acumulado5 <-
rf_triangle_completo5
rf_triangle_acumulado5[,-1] <-
t(apply(rf_triangle_completo5[,-
1], 1, cumsum))
print(rf_triangle_acumulado5)
```

VPF RF

```{r}
#vector de pagos futuros
vpf.rf5 <- rep(0, 9 - 1)
for (k in 1:8) {
  future <-
row(rf_triangle_completo5) +
col(rf_triangle_completo5) - 1
== 9 + k
  vpf.rf5[k] <-
sum(rf_triangle_completo5[futu
re])
}
vpf_RF5<-vpf.rf5;vpf_RF5
```

Pagos futuros RF

```{r}
PF_RF5<-
sum(vpf_RF5);PF_RF5
```
*Renta Vencida ETI a 5 de
mayo de 2025*

```{r}
ETI.V<-
c(0.02119,0.02045,0.02118,0.02
199,0.02275,0.02353,0.02428,0.
02496)
```

```{r}
i.renta5<-
numeric(length(ETI.V))
for (i in 1:length(ETI.V))
{i.renta5[i]<- (1+ETI.V[i])^(-
i)}; i.renta5
prov.renta_RF5<-
sum(vpf_RF5*i.renta5);
prov.renta_RF5
```
Renta vencida

```{r}
prov.renta_RF5<-
sum(vpf_RF5*i.renta5);
prov.renta_RF5
```


LASSO

```{r}
train_sample5_lasso<-
train_sample5
test_sample5_lasso<-
test_sample5
```

```{r}
X_train <- model.matrix(
  Paid ~ AccYear + DevYear +
Type + Age + AccMonth +
AccWeekday + RepDelDays +
Status,
  data = train_sample5_lasso)[, -
1]
y_train <-
train_sample5_lasso$Paid

X  test <- model.matrix(
  Paid ~ AccYear + DevYear +
Type + Age + AccMonth +
```

```
AccWeekday + RepDelDays + Status,
  data = test_sample5_lasso)[, -1]
y_test <- test_sample5_lasso$Paid

set.seed(123)
cv_lasso <- cv.glmnet(X_train, y_train, alpha = 1, nfolds = 5) # alpha=1 para Lasso

# Lambda óptimo
lambda_opt <- cv_lasso$lambda.min

# Ajusta el modelo final
lasso_model <- glmnet(X_train, y_train, alpha = 1, lambda = lambda_opt)

coef_lasso <- coef(lasso_model)
print(coef_lasso)
# Las variables con coeficiente distinto de cero son las seleccionadas por Lasso

```


PREDICCIÓN DEL ERROR (TEST)

PREDICCIÓN

```{r}
# Predice pagos incrementales para el test
test_sample5_lasso$Paid_pred_lasso <- predict(lasso_model, newx = X_test, s = lambda_opt)
```

Cálculo del RMSE y NRMSE

```{r}
# RMSE
rmse_lasso <- sqrt(mean((y_test - test_sample5_lasso$Paid_pred_lasso)^2, na.rm = TRUE))
# rNMSE relativo a la media
rnmse_lasso <- rmse_lasso / mean(y_test, na.rm = TRUE)
# NRMSE por rango
nrmse_lasso <- rmse_lasso / (max(y_test, na.rm = TRUE) - min(y_test, na.rm = TRUE))

cat("RMSE:", rmse_lasso, "\n")
cat("rNMSE (media):", rnmse_lasso, "\n")
cat("NRMSE (rango):", nrmse_lasso, "\n")
```

BBDD Pagos futuros

```{r}
# Para cada AccYear, identificar el máximo DevYear con datos reales
lasso_max_dev_by_accyear5 <- panel5 %>%
  filter(AccYear <= 2021) %>%
  group_by(AccYear) %>%
  summarise(max_dev = max(DevYear[Paid > 0], na.rm = TRUE))
```

```{r}
# Unión del máximo DevYear al panel
lasso_panel_futuro5 <- panel5 %>%
  filter(AccYear <= 2021) %>%

left_join(lasso_max_dev_by_accyear5, by = "AccYear") %>%
  filter(DevYear > max  dev)
```

```{r}
table(lasso_panel_futuro5$AccYear)
```

Predicción de pagos incrementales futuros

```{r}
X_futuro <- model.matrix(
  Paid ~ AccYear + DevYear + Type + Age + AccMonth + AccWeekday + RepDelDays + Status,
  data = lasso_panel_futuro5)[, -1]

lasso_panel_futuro5$Paid_pred_lasso <- predict(lasso_model, newx = X_futuro, s = lambda  opt)
lasso_panel_futuro5$Paid_pred_lasso <- pmax(lasso_panel_futuro5$Paid_pred_lasso, 0)
```

Construcción de triángulos

Triángulo real

```{r}
# Triángulo real (pagos históricos)
triangle_real5 <- panel5 %>%
  filter(AccYear <= 2021) %>%
  group_by(AccYear, DevYear) %>%
  summarise(Paid = sum(Paid, na.rm = TRUE), .groups = "drop")

triangle_real_wide5 <- triangle_real5 %>%
  pivot_wider(
    names_from = DevYear,
    values_from = Paid,
    names_sort = TRUE )

print(triangle_real_wide5/1000)
```

Triángulo futuro

```{r}
#Triángulo futuro (pagos predichos)
lasso_triangle_pred_futuro5 <- lasso_panel_futuro5 %>%
  group_by(AccYear, DevYear) %>%
  summarise(Paid = sum(Paid_pred_lasso, na.rm = TRUE), .groups = "drop") %>%
  tidyr::spread(key = DevYear, value = Paid)

print(lasso_triangle_pred_futuro5)

lasso_triangle_pred_futuro5_long <- lasso_triangle_pred_futuro5 %>%
  pivot_longer(
    cols = -AccYear,
    names_to = "DevYear",
    values_to = "Paid"
  ) %>%
  mutate(DevYear = as.numeric(DevYear))

# Gráfico de líneas por año de ocurrencia
ggplot(lasso_triangle_pred_futuro5_long, aes(x = DevYear, y = Paid, color = as.factor(AccYear))) +
  geom_line() +
  geom  point() +
  labs(
    title = "Pagos predichos por año de desarrollo (Lasso)",
    x = "Año de desarrollo",
    y = "Pago predicho",
    color = "Año de ocurrencia"
  ) +
  theme_minimal()

```

Triángulo combinado (real + predicho)

```{r}
```

```
# Triángulo futuro (pagos predichos, ya truncados a cero)
lasso_triangle_futuro5 <- lasso_panel_futuro5 %>%
  group_by(AccYear, DevYear) %>%
  summarise(Paid = sum(Paid_pred_lasso, na.rm = TRUE), .groups = "drop")

# Combina ambos: si hay predicción, úsala; si no, usa el valor real
lasso_triangle_completo5 <- bind_rows(triangle_real5, lasso_triangle_futuro5) %>%
  group_by(AccYear, DevYear) %>%
  summarise(Paid = sum(Paid, na.rm = TRUE), .groups = "drop") %>%
  tidyr::spread(key = DevYear, value = Paid)

lasso_triangle_completo5<- lasso_triangle_completo5/1000
#-Triangulo individuales por año de desarrollo, NO acumulados
print(lasso_triangle_completo5)
#Desacumulado
```

Triangulo acumulado

```{r}
lasso_triangle_acumulado5 <- lasso_triangle_completo5
lasso_triangle_acumulado5[,-1] <- t(apply(lasso_triangle_acumulado5[,-1], 1, cumsum))
print(lasso_triangle_acumulado5)
```

VPF LASSO

```{r}
#vector de pagos futuros
vpf.lasso5 <- rep(0, 9 - 1)
for (k in 1:9 - 1) {
  future <- row(lasso_triangle_completo5) + col(lasso_triangle_completo5) - 1 == 9 + k
  vpf.lasso5[k] <- sum(lasso_triangle_completo5[future])
}
vpf_LASSO5<- vpf.lasso5;vpf_LASSO5
```

Pagos futuros LASSO

```{r}
PF_LASSO5<- sum(vpf_LASSO5);PF_LASSO5
```

Renta Vencida ETI a 5 de mayo de 2025

```{r}
ETI.V<- c(0.02119,0.02045,0.02118,0.02199,0.02275,0.02353,0.02428,0.02496)
```

```{r}
i.renta5<- numeric(length(ETI.V))
for (i in 1:length(ETI.V)) {i.renta5[i]<- (1+ETI.V[i])^(-i)}; i.renta5
prov.renta_LASSO5<- sum(vpf_LASSO5*i.renta5); prov.renta_LASSO5
```

Renta vencida

```{r}
prov.renta_LASSO5<- sum(vpf_LASSO5*i.renta5); prov.renta_LASSO5
```

GLM

```{r}
train_sample5_glm<- train_sample5
test_sample5_glm<- test_sample5
```

```{r}
# Entrena el modelo GLM Poisson con link log
model_glm5 <- glm(
  Paid ~ AccYear + DevYear + Type + Age + AccMonth + AccWeekday + RepDelDays + Status,
  data = train_sample5_glm,
  family = poisson(link = "log"))
summary(model_glm5)
```

PREDICCIÓN DEL ERROR (TEST)

PREDICCIÓN

```{r}
# Predice pagos incrementales para el test
test_sample5_glm$Paid_pred_glm <- predict(model_glm5, newdata = test_sample5_glm, type = "response")
```

Cálculo del RMSE y NRMSE

```{r}
# RMSE
rmse_glm <- sqrt(mean((test_sample5_glm$Paid - test_sample5_glm$Paid_pred_glm)^2, na.rm = TRUE))
# rNMSE relativo a la media
rnmse_glm <- rmse_glm / mean(test_sample5_glm$Paid, na.rm = TRUE)
# NRMSE por rango
nrmse_glm <- rmse_glm / (max(test_sample5_glm$Paid, na.rm = TRUE) - min(test  sample5  glm$Paid, na.rm = TRUE))

cat("RMSE:", rmse_glm, "\n")
cat("rNMSE (media):", rnmse_glm, "\n")
cat("NRMSE (rango):", nrmse_glm, "\n")
```

BBDD Pagos futuros

```{r}
# Para cada AccYear, identificar el máximo DevYear con datos reales
glm_max_dev_by_accyear5 <- panel5 %>%
  filter(AccYear <= 2021) %>%
  group_by(AccYear) %>%
  summarise(max_dev = max(DevYear[Paid > 0], na.rm = TRUE))

# Unión del máximo DevYear al panel
glm_panel_futuro5 <- panel5 %>%
  filter(AccYear <= 2021) %>%

left_join(glm_max_dev_by_accyear5, by = "AccYear") %>%
  filter(DevYear > max_dev)  # Solo los años de desarrollo futuros (incompletos)
```

```{r}
table(glm  panel  futuro5$AccYear)
```

Predicción de pagos incrementales futuros

```{r}
glm_panel_futuro5$Paid_pred_glm <- predict(model_glm5, newdata = glm_panel_futuro5, type = "response")
glm_panel_futuro5$Paid_pred_glm <- pmax(glm_panel_futuro5$Paid_pred_glm, 0)
```

Construcción de triángulos

Triángulo real

```{r}
# Triángulo real (pagos históricos)
triangle  real5 <- panel5 %>%
  filter(AccYear <= 2021) %>%
  group_by(AccYear, DevYear) %>%
  summarise(Paid = sum(Paid, na.rm = TRUE), .groups = "drop")

triangle_real_wide5 <- triangle_real5 %>%
  pivot_wider(
    names_from = DevYear,
    values_from = Paid,
    names_sort = TRUE
  )

print(triangle_real_wide5/1000)
```

Triángulo futuro

```{r}
#Triángulo futuro (pagos predichos)
glm_triangle_pred_futuro5 <- glm_panel_futuro5 %>%
  group_by(AccYear, DevYear) %>%
  summarise(Paid = sum(Paid_pred_glm, na.rm = TRUE), .groups = "drop") %>%
  tidyr::spread(key = DevYear, value = Paid)

print(glm_triangle_pred_futuro5)

glm_triangle_pred_futuro5_long <- glm_triangle_pred_futuro5 %>%
  pivot_longer(
    cols = -AccYear,
    names_to = "DevYear",
    values_to = "Paid"
  ) %>%
  mutate(DevYear = as.numeric(DevYear))

# Gráfico de líneas por año de ocurrencia
ggplot(glm_triangle_pred_futuro5_long, aes(x = DevYear, y = Paid, color = as.factor(AccYear))) +
  geom_line() +
  geom_point() +
  labs(
    title = "Pagos predichos por año de desarrollo (GLM)",
    x = "Año de desarrollo",
    y = "Pago predicho",
    color = "Año de ocurrencia"
  ) +
  theme_minimal()
```

Triángulo combinado (real + predicho)

```{r}
glm_triangle_futuro5 <- glm_panel_futuro5 %>%
  group_by(AccYear, DevYear) %>%
  summarise(Paid = sum(Paid_pred_glm, na.rm = TRUE), .groups = "drop")

glm_triangle_completo5 <- bind_rows(triangle_real5, glm_triangle_futuro5) %>%
  group_by(AccYear, DevYear) %>%
  summarise(Paid = sum(Paid, na.rm = TRUE), .groups = "drop") %>%
  tidyr::spread(key = DevYear, value = Paid)

glm_triangle_completo5 <- glm_triangle_completo5/1000
print(glm_triangle_completo5)

write.csv(glm_triangle_completo5, file = "GLM_ML_Triangulo_desacumulado", row.names = FALSE)
```

Triangulo acumulado

```{r}
glm_triangle_acumulado5 <- glm_triangle_completo5
glm_triangle_acumulado5[,-1] <- t(apply(glm_triangle_acumulado5[,-1], 1, cumsum))
print(glm_triangle_acumulado5)

```

VPF GLM

```{r}
# Vector de pagos futuros
vpf.glm5 <- rep(0, 9 - 1)
for (k in 1:8) {
  future <- row(glm_triangle_completo5) + col(glm_triangle_completo5) - 1 == 9 + k
  vpf.glm5[k] <- sum(glm_triangle_completo5[future])
}
vpf_GLM5 <- vpf.glm5; vpf_GLM5

# Pagos futuros totales
PF_GLM5 <- sum(vpf_GLM5); PF_GLM5

# Renta vencida ETI a 5 de mayo de 2025
ETI.V <- c(0.02119,0.02045,0.02118,0.02199,0.02275,0.02353,0.02428,0.02496)
i.renta5 <- numeric(length(ETI.V))
for (i in 1:length(ETI.V)) {i.renta5[i] <- (1+ETI.V[i])^(-i)}
prov.renta_GLM5 <- sum(vpf_GLM5 * i.renta5); prov.renta_GLM5

```

KNN

```{r}
library(FNN)


# Selecciona variables y elimina NAs
train_knn <- train_sample5 %>%
  select(Paid, AccYear, DevYear, Type, Age, AccMonth, AccWeekday, RepDelDays, Status) %>%
  na.omit()

test_knn <- test_sample5 %>%
  select(Paid, AccYear, DevYear, Type, Age, AccMonth, AccWeekday, RepDelDays, Status) %>%
  na.omit()

# Convierte factores a variables dummy
train_knn_mat <- model.matrix(
  Paid ~ AccYear + DevYear + Type + Age + AccMonth +
```

```
AccWeekday + RepDelDays +
Status,
  data = train_knn
)[, -1]
y_train <- train_knn$Paid

test_knn_mat <- model.matrix(
  Paid ~ AccYear + DevYear +
Type + Age + AccMonth +
AccWeekday + RepDelDays +
Status,
  data = test_knn
)[, -1]
y_test <- test_knn$Paid
```

PREDICCIÓN DEL ERROR (TEST)

PREDICCIÓN

```{r}
set.seed(123)
k <- 5
knn_pred <- knn.reg(
  train = train_knn_mat,
  test = test_knn_mat,
  y = y_train,
  k = k
)$pred

test_knn$Paid_pred_knn <-
pmax(knn_pred, 0)
```

Cálculo del RMSE y NRMSE

```{r}
rmse_knn <- sqrt(mean((y_test -
test_knn$Paid_pred_knn)^2,
na.rm = TRUE))
rnmse_knn <- rmse_knn /
mean(y_test, na.rm = TRUE)
nrmse_knn <- rmse_knn /
(max(y_test, na.rm = TRUE) -
min(y  test, na.rm = TRUE))

cat("RMSE:", rmse_knn, "\n")
cat("rNMSE (media):",
rnmse_knn, "\n")
cat("NRMSE (rango):",
nrmse_knn, "\n")
```

BBDD Pagos futuros

```{r}
# Para cada AccYear, identificar
el máximo DevYear con datos
reales
knn_max_dev_by_accyear5 <-
panel5 %>%
  filter(AccYear <= 2021) %>%
  group_by(AccYear) %>%
  summarise(max_dev =
max(DevYear[Paid > 0], na.rm
= TRUE))

# Unión del máximo DevYear al
panel
knn_panel_futuro5 <- panel5
%>%
  filter(AccYear <= 2021) %>%

left_join(knn_max_dev_by_acc
year5, by = "AccYear") %>%
  filter(DevYear > max_dev)

table(knn_panel_futuro5$AccY
ear)
```

Predicción de pagos incrementales futuros

```{r}
futuro_knn_mat <-
model.matrix(
  Paid ~ AccYear + DevYear +
Type + Age + AccMonth +
AccWeekday + RepDelDays +
Status,
  data = knn_panel_futuro5
)[, -1]

futuro_knn_pred <- knn.reg(
  train = train_knn_mat,
  test = futuro_knn_mat,
  y = y_train,
  k = k
)$pred

knn_panel_futuro5$Paid_pred_
knn <- pmax(futuro_knn_pred,
0)
```

Construcción de triángulos

Triángulo real

```{r}
# Triángulo real (pagos
históricos)
triangle  real5 <- panel5 %>%
  filter(AccYear <= 2021) %>%
  group_by(AccYear, DevYear)
%>%
  summarise(Paid = sum(Paid,
na.rm = TRUE), .groups =
"drop")

triangle_real_wide5 <-
triangle_real5 %>%
  pivot_wider(
    names_from = DevYear,
    values_from = Paid,
    names  sort = TRUE
  )
print(triangle_real_wide5/1000)
```

Triángulo futuro

```{r}
#Triángulo futuro (pagos
predichos)
knn_triangle_pred_futuro5 <-
knn_panel_futuro5 %>%
  group_by(AccYear, DevYear)
%>%
  summarise(Paid =
sum(Paid_pred_knn, na.rm =
TRUE), .groups = "drop") %>%
  tidyr::spread(key = DevYear,
value = Paid)

print(knn_triangle_pred_futuro5
)
```

```{r}
knn triangle pred futuro5 long
<- knn_triangle_pred_futuro5
%>%
  pivot_longer(
    cols = -AccYear,
    names_to = "DevYear",
    values_to = "Paid"
  ) %>%
  mutate(DevYear =
as.numeric(DevYear))

# Gráfico de líneas por año de
ocurrencia
ggplot(knn_triangle_pred_futur
o5_long, aes(x = DevYear, y =
Paid, color =
as.factor(AccYear))) +
  geom_line() +
  geom_point() +
  labs(
    title = "Pagos predichos por
año de desarrollo (KNN)",
    x = "Año de desarrollo",
    y = "Pago predicho",
    color = "Año de ocurrencia"
  ) +
  theme_minimal()
```

Triángulo combinado (real + predicho)

```{r}
knn_triangle_futuro5 <-
knn_panel_futuro5 %>%
  group_by(AccYear, DevYear)
%>%
  summarise(Paid =
sum(Paid_pred_knn, na.rm =
TRUE), .groups = "drop")

knn_triangle_completo5 <-
bind_rows(triangle_real5,
knn_triangle_futuro5) %>%
```

```
  group_by(AccYear, DevYear)
%>%
  summarise(Paid = sum(Paid,
na.rm = TRUE), .groups =
"drop") %>%
  tidyr::spread(key = DevYear,
value = Paid)

knn_triangle_completo5 <-
knn_triangle_completo5/1000
print(knn_triangle_completo5)

write.csv(knn_triangle_complet
o5, file =
"KNN_Triangulo_desacumulad
o", row.names = FALSE)
```

Triangulo acumulado

```{r}
knn_triangle_acumulado5 <-
knn_triangle_completo5
knn_triangle_acumulado5[,-1]
<-
t(apply(knn_triangle_acumulado
5[,-1], 1, cumsum))
print(knn_triangle_acumulado5)
```

VPF KNN

```{r}
# Vector de pagos futuros
vpf.knn5 <- rep(0, 9 - 1)
for (k in 1:8) {
  future <-
row(knn_triangle_completo5) +
col(knn_triangle_completo5) - 1
== 9 + k
  vpf.knn5[k] <-
sum(knn_triangle_completo5[fu
ture])
}
vpf_KNN5 <- vpf.knn5;
vpf_KNN5

# Pagos futuros totales
PF_KNN5 <- sum(vpf_KNN5);
PF_KNN5

# Renta vencida ETI a 5 de
mayo de 2025
ETI.V <-
c(0.02119,0.02045,0.02118,0.02
199,0.02275,0.02353,0.02428,0.
02496)
i.renta5 <-
numeric(length(ETI.V))
for (i in 1:length(ETI.V))
{i.renta5[i] <- (1+ETI.V[i])^(-
i)}
prov.renta_KNN5 <-
sum(vpf_KNN5 * i.renta5);
prov.renta  KNN5
```

MÉTODOS TRADICIONALES

```{r}

# Lista de vectores, uno por
cada AccYear
triangle_vectors <-
lapply(1:nrow(triangle_real_wid
e5), function(i)
as.numeric(triangle_real_wide5[
i, -1]))
names(triangle_vectors) <-
triangle_real_wide5$AccYear

# Datos del triángulo de
siniestros,
# Años de origen (filas) y años
de desarrollo (columnas)
c0 =
triangle  vectors[["2013"]];c0<-
c0[c(1:9)];c0
c1 =
triangle_vectors[["2014"]];c1<-
c1[c(1:8)];c1
c2 =
triangle_vectors[["2015"]];c2<-
c2[c(1:7)];c2
c3 =
triangle_vectors[["2016"]];c3<-
c3[c(1:6)];c3
c4 =
triangle_vectors[["2017"]];c4<-
c4[c(1:5)];c4
c5 =
triangle_vectors[["2018"]];c5<-
c5[c(1:4)];c5
c6 =
triangle_vectors[["2019"]];c6<-
c6[c(1:3)];c6
c7 =
triangle_vectors[["2020"]];c7<-
c7[c(1:2)];c7
c8 =
triangle_vectors[["2021"]];c8<-
c8[1];c8

C0 <- cumsum(c0);C0
C1 <- c(cumsum(c1),NA);C1
C2 <-
c(cumsum(c2),NA,NA);C2
C3 <-
c(cumsum(c3),NA,NA,NA);C3
C4 <-
c(cumsum(c4),NA,NA,NA,NA)
;C4
C5 <-
c(cumsum(c5),NA,NA,NA,NA,
NA);C5
C6 <-
c(cumsum(c6),NA,NA,NA,NA,
NA,NA);C6
C7 <-
c(cumsum(c7),NA,NA,NA,NA,
NA,NA,NA);C7
C8 <-
c(cumsum(c8),NA,NA,NA,NA,
NA,NA,NA,NA);C8

C <-
matrix(c(C0,C1,C2,C3,C4,C5,C
6,C7,C8),nrow=9,ncol=9)
C <- t(C);
C <- as.triangle(C);
C<-C/1000;C
```

Método Mack Chain-Ladder
Estan /1000

```{r}
library(ChainLadder)
mch <- MackChainLadder(C);
mch
names(mch)
```

```{r}
# Triángulo completo estimado
con Mack
triangle_mack_completo =
mch$FullTriangle
triangle_mack_completo
#Acumulado
```

Triangulo completo

```{r}
mch$FullTriangle # Triangulo
completo
mch$f # Factor de desarrollo
```

VPF Mack

```{r}
# En FullTriangle está el
acumulado, lo desacumulamos
para obtener el VPF
a<-
matrix(c(rep(0,dim(C)[1]),mch$
FullTriangle),nrow=dim(C)[1],n
col=dim(C)[1]);a
noncumFullTriangle<-
mch$FullTriangle-a;
noncumFullTriangle  #
Desacumulado

write.csv(noncumFullTriangle,
file =
"MM_MT_Triangulo_desacum
ulado", row.names = FALSE)
```

Vector de pagos futuros

```{r}
```

```
vpf.mm <- rep(0, dim(C)[1] - 1)
#me va a hacer un vector de 0
con dim k
for (k in 1:dim(C)[1] - 1) {
  future <-
row(noncumFullTriangle) +
col(noncumFullTriangle) - 1 ==
dim(C)[1] + k
  vpf.mm[k] <-
sum(noncumFullTriangle[future
])
}
vpf_MM<-vpf.mm;vpf_MM
```

Pagos futuros Mack

```{r}
PF_MM<-
sum(vpf_MM);PF_MM
```
*Renta Vencida ETI a 5 de
mayo de 2025*

```{r}
ETI.V<-
c(0.02119,0.02045,0.02118,0.02
199,0.02275,0.02353,0.02428,0.
02496)
```

```{r}
i.renta<-numeric(length(ETI.V))
for (i in 1:length(ETI.V))
{i.renta[i]<- (1+ETI.V[i])^(-i)};
i.renta
prov.renta_MM<-
sum(vpf_MM*i.renta);
prov.renta_MM
```
Renta vencida
```{r}
prov.renta_MM<-
sum(vpf_MM*i.renta);
prov.renta_MM
```

Modelo lineal generalizado
con distribución Poisson
(sobredispersa) y función de
enlace logarítmica

```{r}
glmformula<-
glmReserve(C,var.power = 1,
link.power = 0, mse.method =
"formula"); glmformula

names(glmformula)
glmformula$FullTriangle
glmformula$model

Cincr <- cum2incr(C)
Ccum <- incr2cum(Cincr)
```

```
# Lo mismo para el resultado
glmformula$FullTriangle:
noncumFullTriangle <-
cum2incr(glmformula$FullTrian
gle); noncumFullTriangle
write.csv(noncumFullTriangle,
file =
"GLM_MT_Triangulo_desacum
ulado", row.names = FALSE)
vpf.mlg.mt <- rep(0, dim(C)[1] -
1)
for (k in 1:dim(C)[1] - 1) {
future <-
row(noncumFullTriangle) +
col(noncumFullTriangle) - 1 ==
dim(C)[1] + k
  vpf.mlg.mt[k] <-
sum(noncumFullTriangle[future
])
}
vpf_MLG_MT<-vpf.mlg.mt;
vpf_MLG_MT
```

Pagos futuros
```{r}
PF_MLG_MT<-
sum(vpf_MLG_MT);PF_MLG_
MT
```

Renta vencida
```{r}
prov.renta_MLG_MT<-
sum(vpf_MLG_MT*i.renta);
prov.renta_MLG_MT
```

COMPARACIÓN DE
MÉTODOS

```{r}
comparativa <- data.frame(
  Método = c(
    "MT Mack CL",
    "MT GLM (Poisson)",
    "ML Random Forest",
    "ML Lasso",
    "ML GLM",
    "ML KNN"
  ),
  PagosFuturos = c(
    PF_MM,         # resultado
método Mack
    PF_MLG_MT,      # resultado
GLM Poisson tradicional
    PF_RF5,          # resultado
Random Forest
    PF_LASSO5,       # resultado
Lasso
    PF_GLM5,         # resultado
GLM ML
    PF  KNN5         # resultado
KNN
  ),
  ProvisiónRentaVencida = c(
    prov.renta_MM,      #
provisión Mack
    prov.renta_MLG_MT,   #
provisión GLM Poisson
tradicional
    prov.renta_RF5,       #
provisión Random Forest
    prov.renta_LASSO5,     #
provisión Lasso
    prov.renta_GLM5,       #
provisión GLM ML
    prov.renta_KNN5        #
provisión KNN
  )
)

print(comparativa)

write.csv(comparativa, file =
"comparativa.csv", row.names =
FALSE)

```

```{r}
comparativa_errores <-
data.frame(
  Método = c(
    "Random Forest",
    "Lasso",
    "GLM ML",
    "KNN"
  ),
  RMSE = c(
    RMSE_RF,        # RMSE
Random Forest
    rmse_lasso,     # RMSE Lasso
    rmse_glm,       # RMSE GLM
ML
    rmse_knn        # RMSE KNN
  ),
  NRMSE = c(
    NRMSE_RF,       # NRMSE
Random Forest
    nrmse_lasso,    # NRMSE
Lasso
    nrmse_glm,      # NRMSE
GLM ML
    nrmse_knn       # NRMSE
KNN
  )
)

print(comparativa_errores)
write.csv(comparativa_errores,
file = "comparativa_errores",
row.names = FALSE)

```

```{r}
library(ggplot2)
library(tidyr)

comparativa_long <-
pivot_longer(
```

```
  comparativa,
  cols = c("PagosFuturos",
"ProvisiónRentaVencida"),
  names_to = "Indicador",
  values_to = "Valor"
)

colores_azules <-
c("PagosFuturos" = "#377eb8",

"ProvisiónRentaVencida" =
"#4daf4a")

colores_azules <-
c("PagosFuturos" = "#1f78b4",

"ProvisiónRentaVencida" =
"#a6cee3")

g<-ggplot(comparativa_long,
aes(x = Método, y = Valor, fill =
Indicador)) +
  geom_bar(stat = "identity",
position = position  dodge()) +
  scale_fill_manual(values =
colores_azules,
                    labels = c("Pagos
Futuros", "Provisión Renta
Vencida")) +
  labs(
    title = "Comparativa de
Métodos: Pagos Futuros y
Provisión Renta Vencida",
    x = "Método",
    y = "Euros",
    fill = "Indicador"
  ) +
  theme_minimal() +
  theme(axis.text.x =
element_text(angle = 25, hjust =
1))
ggsave("comparativa_metodos.p
df", plot = g, width = 9, height =
5)
g
```