

Proyecto Modelos II:
SUPPORT2

Miguel Angel Puerta Vasquez
ID : 1000760164
Leon Mateo Velez Gonzáles
ID: 1216728793

Universidad De Antioquia
Modelos y Simulación II
Julian David Arias Londoño
18 de Noviembre de 2025

En la evaluación de la supervivencia de pacientes en estado crítico es fundamental analizar las condiciones de mejoría que presentan bajo cuidados médicos. Para ello, es necesario considerar sus características fisiológicas, demográficas, sociales y funcionales, así como la severidad de su enfermedad y la presencia de comorbilidades previas.

Con el análisis de todos estos datos es posible estimar el tiempo de supervivencia del paciente y determinar si resulta adecuado prolongar o no las intervenciones médicas cuando la probabilidad de supervivencia es baja.

Dado el gran volumen de información y la diversidad de factores que influyen en el pronóstico, una solución basada en Machine Learning puede facilitar la estimación de la supervivencia y permitir una identificación más rápida de los pacientes con mejores probabilidades, contribuyendo así a una atención médica más efectiva.

Para este desarrollo se hará uso del dataset SUPPORT2, que es un dataset enfocado a la predicción de la tasa de supervivencia dentro de 2 meses y 6 meses de pacientes en estado crítico, basado en sus condiciones de vida, enfermedades, reportes clínicos y otra información, este posee una base de datos con 9105 muestras, cada una de estas asociada a un paciente, y 47 variables, de estas 47 variables se tiene:

- **Información demográfica:** age, sex, edu, income, race.
- **Variables relacionadas con el tipo de enfermedad:** dzgroup, dzclass, ca.
- **Variables clínicas:** meanbp, wblc, hrt, resp, temp, pafi, alb, bili, crea, sod, ph, glucose, bun, urine.
- **Variables de estado funcional:** adlp, adls, sfdm2, adlsc.
- **Comorbilidades:** num.co, diabetes, dementia.
- **Decisiones Médicas:** prg2m, prg6m, dnr, dnrday.
- **Variables de costes:** charges, totcst, totmcst, avtisst.
- **Severidad fisiológica:** scoma, sps, aps.
- **Variables de tiempo:** d.time, slos, hday.
- **Estado de vida:** death, hospdead.
- **Variables a predecir:** surv2m, surv6m.

Nuestro dataset posee alta cantidad de datos faltantes, como tal se deberá realizar imputación para rellenar los datos faltantes, algunos de estos son recomendados por el dataset en si mismo ya que dan buenos resultados, como lo son:

- | | |
|----------------------|----------------------|
| - alb: 3.5 | - bun: 6.51 |
| - pafi: 333.3 | - wblc: 9 |
| - bili: 1.01 | - urine: 2502 |
| - crea: 1.01 | |

Eso nos deja con los siguientes valores faltantes, a los cuales les realizamos la siguiente imputación:

- **Mediana:**

edu (12), scoma (0), charges (25024), totcst (14452.73), totmcst (13223.5), avtisst (19.5), sps (23.90), aps (34), prg2m (0.7), prg6m (0.5), meanbp (77), hrt (100), resp (24), temp (36.69), sod (137), glucose (135), adlp (0), adls (1).

- **Moda:**

race: “white” (categoría más frecuente).

dnr: se asigna “no DNR” (valor dominante).

surv2m y surv6m: 0.000000 (valor predominante; solo un faltante en cada caso).

- **Imputaciones específicas:**

income: categoría “11–25k”, usada como valor representativo.

ph: promedio general (7.415), debido a su variabilidad fisiológica.

dnrday: 0, consistente con casos faltantes coincidentes con dnr.

sfdm2: 159 pacientes imputados con “<2 mo follow-up” y el resto con “no (Month 2 and SIP pres)”, siguiendo el patrón descrito en el dataset.

Para la codificación de variables, se realizará de la siguiente manera:

- **Númérica (StandardScaler):**

age, meanbp, wblc, hrt, resp, temp, pafi, alb, bili, crea, sod, ph, glucose, bun, urine, adlsc, dnrday, charges, totcst, totmcst, avtisst, scoma, sps, aps.

- **Binary Encoding:**

sex, diabetes, dementia, death, hospdead.

- **Label Encoding:**

income, ca.

- **One-Hot Encoding:**

race, dzgroup, dzclass, sfdm2, dnr.

- **No se realiza codificación:**

edu, adlp, adls, num.co, prg2m, prg6m, d.time, slos, hday, surv2m, surv6m

Se hará uso de un paradigma de aprendizaje supervisado, ya que el objetivo está claramente definido para los pacientes, que es su probabilidad de fallecimiento en 2 y 6 meses. El preprocesamiento ayudará a construir un dataset que permita aplicar algoritmos estándar de predicción como lo son Random Forest, SVC, entre otros algoritmos.

Learning Patient-Specific Cancer Survival Distributions as a Sequence of Dependent Regressors

En este se emplea un paradigma de aprendizaje supervisado para el análisis de supervivencia, usando como técnica principal el modelo Multi-Task Logistic Regression. La validación se realiza mediante particionado train/test y comparación directa con modelos clásicos como Cox y Aalen. Se evalúa el desempeño con MSE de probabilidad en distintos horizontes de tiempo, además de errores como AE, AE-log y RAE. Los resultados muestran que el MTLR obtiene mejores resultados, ya que obtiene menor MSE y error comparado con otros métodos, demostrando que es un buen método de predicción de tiempos de vida.

AE = $|y_i - \hat{y}_i|$, error absoluto entre predicción y valor real.

AE-log = $|\log(y_i + 1) - \log(\hat{y}_i + 1)|$, usado cuando la escala del error debe penalizar más valores pequeños.

RAE = $\frac{\sum_{i=1}^n |y_i - \hat{y}_i|}{\sum_{i=1}^n |y_i - \bar{y}_i|}$, error absoluto relativo

comparado con un modelo base

Personalized Survival Prediction with Contextual Explanation Networks

Se emplea un paradigma de aprendizaje supervisado para análisis de supervivencia, reformulando la predicción temporal como un problema de clasificación secuencial por intervalos, usando como técnica principal la Contextual Explanation Networks, este es comparado con Cox, Aalen, CRF, MLP-CRF y LSTM-CRF. La validación se realiza mediante train/validation/test fijo y 5-fold cross-validation. Se utiliza Acc@K y RAE para evaluar el desempeño. Los resultados obtenidos muestran que los métodos basados en CEN superan a los modelos clásicos, como Cox y Aalen, y variantes como MLP-CRF/LSTM-CRF, destacándose LSTM-CEN como el de mejor rendimiento entre los métodos comparados.

Acc@K = $\frac{1}{n} \sum_{i=1}^n 1\{y_i \in \text{Top-K}(\hat{y}_i)\}$, acierto si el ítem correcto está en el top-K.

DeepSurv: personalized treatment recommender system using a Cox proportional hazards deep neural network

Emplean un paradigma de aprendizaje supervisado para el análisis de supervivencia, usando como técnica principal DeepSurv, una red neuronal profunda que optimiza el modelo Cox. La validación es realizada mediante un conjunto de prueba independiente y bootstrap para intervalos de confianza. El desempeño es evaluado mediante el uso de C-index. Los resultados muestran que DeepSurv supera al modelo de Cox base y tiene un rendimiento comparable o superior al Random Survival Forest, teniendo un mejor C-index.

C-index = $\frac{1}{n} \sum_{i=1}^n 1[(y_i > y_j) = (\hat{y}_i > \hat{y}_j)]$, Mide cuántos pares ordena bien el modelo.

Let the Experts Speak: Improving Survival Prediction & Calibration via Mixture-of-Experts Heads

Se hace uso de un paradigma de aprendizaje supervisado para análisis de supervivencia, usan como técnica principal el Mixture-of-Experts con redes neuronales y una función de pérdida basada en MTLR. La validación es realizada mediante una división train/validation/test y replica con múltiples semillas para comparar contra otros modelos base como CoxPH, Random Survival Forest y MTLR. El desempeño es evaluado mediante C-index y Brier Score. Los resultados muestran que el modelo Personalized-MoE obtiene el mejor desempeño en SUPPORT2,

BS = $\frac{1}{n} \sum_{i=1}^n (p_i - y_i)^2$, Mide error cuadrático para probabilidad

Para tener una garantía de que los resultados sean reproducibles y evaluados de manera rigurosa, se definirá una metodología de validación que considera las características particulares del dataset, que son la alta dimensionalidad, la presencia de los datos faltantes, posible desequilibrio de clases y un riesgo a sobreajuste. La metodología a adoptar será la siguiente:

- **División del conjunto de datos:**

Se dividirá el conjunto de datos en tres subconjuntos, entrenamiento, validación y test, el subconjunto de entrenamiento tendrá un 70% de los datos, mientras validación y test tendrán 15% cada uno, para dividir correctamente el 100% de los datos.

- **Validación cruzada estratificada:**

Ya que el dataset puede presentar desequilibrio entre las clases de supervivencia vs no supervivencia, y a que algunas configuraciones pueden ser sensibles a la varianza en el muestreo elegido, aplicaremos un esquema de validación cruzada estratificada de k-folds, que usará $k = 5$ folds y estratificando por clase.

- **Pipeline de preprocesamiento:**

Para asegurar de que cada etapa de procesamiento se realice correctamente sin fuga de información, se emplea un pipeline integrado que realiza imputación, escalamiento, codificación de variables categóricas, técnicas de balanceo y entrenamiento del clasificador.

- **Selección de modelo:**

Se realizará una comparación del desempeño promedio de cada algoritmo con la validación cruzada, su estabilidad, evaluando su capacidad de calibración y finalmente se probará sus resultados con el subconjunto de test.

Con el objetivo de obtener un modelo robusto y comparar diferentes enfoques, se seleccionaron cinco algoritmos para comparar sus resultados y evaluar su desempeño:

- Modelo paramétrico: Regresión lineal
- Modelo no paramétrico: K-Nearest Neighbors (KNN)
- Modelo basado en ensamble: Random Forest
- Red neuronal artificial: Multilayer Perceptron Regressor
- Máquina de vectores de soporte: SVR con kernel rbf.

A cada uno de estos modelos se les ha aplicado un ajuste de hiperparametros mediante grid search, para evaluar cuál ajuste es el mejor para cada uno, a continuación veremos en la tabla 1 los hiperparametros que se definieron para cada modelo y su malla de valores usada para cada uno.

Modelo	Tipo	Hiperparámetro	Malla
Regresión Lineal	Paramétrico	fit_intercept	[True, False]
KNN	No Paramétrico	n_neighbors	[3,5,7]
		weights	["uniform","distance"]
		p	[1,2]
Random Forest	Ensamble	n_estimators	[100,200]
		max_depth	[None,5,10]
		min_samples_split	[2, 5]
MLP	Red neuronal	hidden_layer_sizes	[(32,), (64,)]
		activation	["relu","tanh"]
		learning_rate_init	[0.001, 0.01]
SVR	SVM	kernel	["rbf"]
		C	[0.1,1,10]
		gamma	["scale","auto"]

Tabla 1. Modelos y su malla de valores usados para buscar el mejor desempeño.

Para evaluar cuáles de los anteriores modelos y configuración de hiperparámetro tiene un mejor desempeño y resulta el mejor para nuestro desarrollo, usaremos las siguientes métricas de desempeño para evaluar los modelos.

- **Error Absoluto Medio (MAE)**

$$MAE = \frac{1}{n} \sum_{i=1}^n |y_i - \hat{y}_i|$$

Este mide el promedio de la diferencia absoluta entre la predicción y el valor real, indica que tan lejos, en promedio, están las predicciones de la probabilidad real de supervivencia.

- **Error Cuadrático Medio (MSE) o Raíz del Error Cuadrático Medio (RMSE)**

$$MSE = \frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2, \quad RMSE = \sqrt{MSE}$$

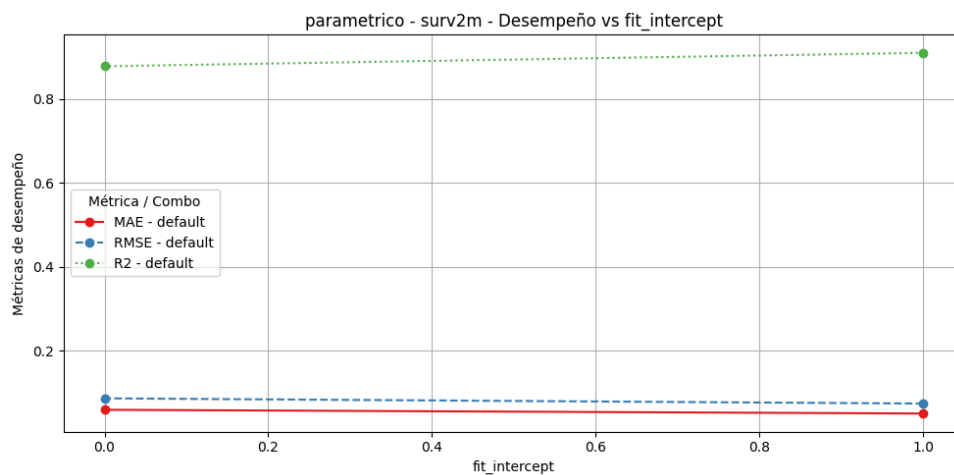
Este penaliza fuertemente los errores grandes, en el caso de la evaluación de modelos de supervivencia probabilística, este puede indicar que el modelo evita predicciones que pueden estar muy alejadas de la realidad.

- R^2 (Coeficiente de Determinación)

$$R^2 = 1 - \frac{\sum_{i=1}^n (y_i - \hat{y}_i)^2}{\sum_{i=1}^n (y_i - \bar{y})^2}$$

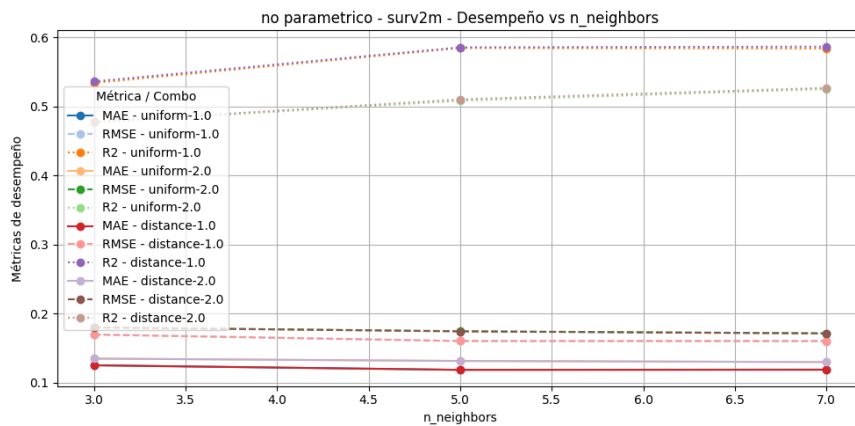
La proporción de varianza explicada por el modelo, este proporciona un valor normalizado entre 0 y 1 de que tanto el modelo captura la variabilidad del objetivo, este nos permite comparar modelos de manera relativa, especialmente útil en nuestro caso que existen varias variables de supervivencia.

Para evaluar los resultados obtenidos por los diferentes modelos y sus configuraciones de hiperparámetros, usaremos el MAE, RMSE y R2 para concluir cuál es mejor de todos los modelos.



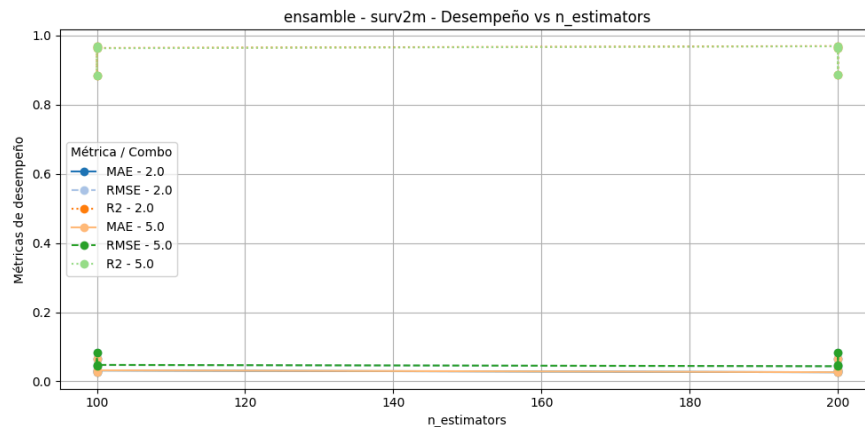
Gráfica 1. Métricas de desempeño contra fit_intercept

En el caso de la *Gráfica 1* para modelos paramétricos podemos notar que el modelo para la mejor predicción de surv2m y surv6m es **fit_intercept = true**, por lo tanto este será el escogido de los dos, esté a su vez tiene un alto coeficiente R2, lo que lleva a que este modelo sea bueno explicando la variación de los datos reales.



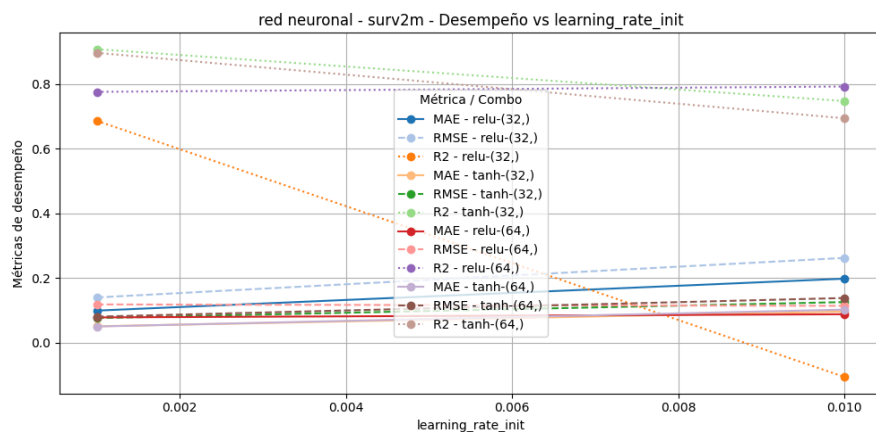
Gráfica 2. Métricas de desempeño contra $n_neighbors$, weights y p

En la *gráfica 2* los modelos no paramétricos demuestran unos resultados decentes pero bastante por debajo de los obtenidos por los modelos paramétricos, se nota una mejoría cuantos más neighbors se usan, y su mejor desempeño ocurre con **weights = distance**, **$p = 1$** , **neighbors = 7**.



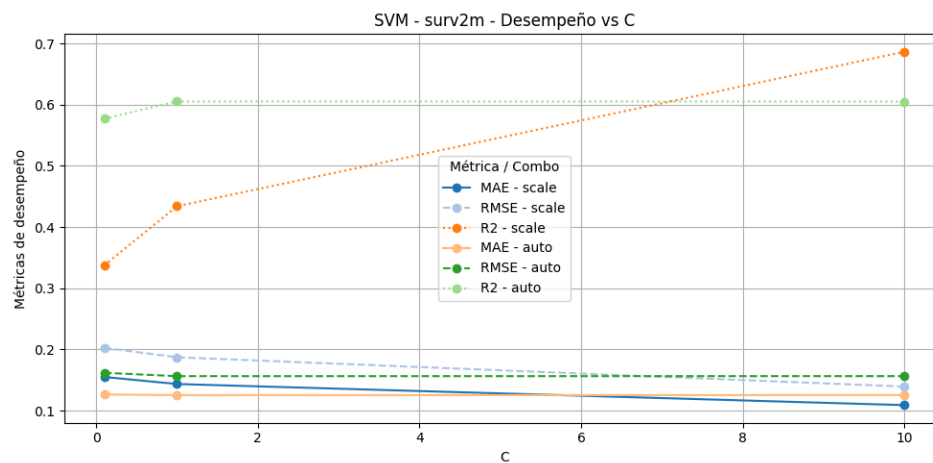
Gráfica 3. Métricas de desempeño contra $n_estimators$ y $min_samples_split$

Los modelos de ensamble tienen un muy buen desempeño y son mejores que los paramétricos como podemos notar en la *gráfica 3* por un buen margen, y su mejor desempeño ocurre con **$n_estimators = 100$** , **$min_samples_split = 2$** , esto puede darse a que incrementar el número de estimadores no garantiza mejoras en el modelo, e incluso podría empeorar ligeramente.



Gráfica 4. Métricas de desempeño contra $learning_rate_init$, $hidden_layer_sizes$ y activation

Mientras en la *gráfica 4* los modelos de redes neuronales tienen un desempeño que varía mucho en sus hiperparametros, y su mejor desempeño ocurre con **learning_rate_init = 0.001**, **activation = tanh** y **hidden_layer_sizes=(32,)**, teniendo resultados cercanos a los obtenidos por el paramétrico, pero un poco inferiores.



Gráfica 5. Métricas de desempeño contra C y gamma

Finalmente la gráfica 5 demuestra que los modelos SVM tienen un desempeño muy decente, solo superando al desempeño de los no paramétricos, pero quedándose por detrás de los demás, y su mejor desempeño ocurre con **C= 10.0** y **gamma = scale**.

Modelo	Objetivo	MAE	RMSE	R2
Paramétrico	surv2m	0.050754 (0.003)	0.074531 (0.009)	0.910362 (0.002)
No paramétrico		0.118614 (0.006)	0.159996 (0.009)	0.586920 (0.005)
Ensamble		0.026321 (0.002)	0.043286 (0.003)	0.969765 (0.005)
Red neuronal		0.050082 (0.003)	0.075592 (0.006)	0.907792 (0.002)
SVM		0.109092 (0.005)	0.139403 (0.007)	0.686409 (0.003)

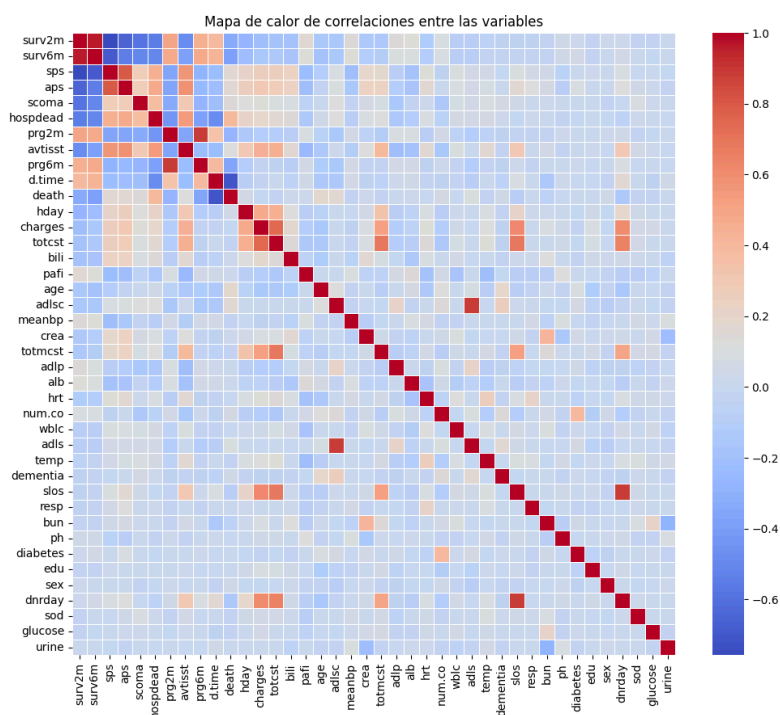
Tabla 2. Comparación MAE, RMSE y R2 para los 5 mejores modelos de cada tipo

Como podemos notar de los modelos de la tabla 2, podemos ver que el modelo de ensamble, es el que tiene los mejores resultados en las variables de desempeño, con el modelo paramétrico siendo el segundo mejor modelo. Con estos datos obtenidos continuaremos trabajando con estos dos modelos principalmente

Variable	Tipo	Spearman	Punto Biserial	p value	Mutual Info
death	Numérica	NaN	NaN	NaN	0.002251
sex	Numérica	NaN	NaN	NaN	0.001757
num.co	Numérica	NaN	NaN	NaN	0.000823
glucose	Numérica	NaN	NaN	NaN	0.000823
urine	Numérica	NaN	NaN	NaN	0.000658
edu	Numérica	NaN	NaN	NaN	0.000604
prg2m	Numérica	NaN	NaN	NaN	0.000329
alb	Numérica	NaN	NaN	NaN	0.000329
resp	Numérica	NaN	NaN	NaN	0.000274

Tabla 3. Top 9 aporte de información de variables a variable objetivo surv2m

Como podemos ver de las 9 variables que más información aportan a la variable objetivo, se puede notar que muy pocas variables influyen de gran manera en nuestra variable a predecir, pero esto no es suficiente para tomar una decisión, los demás valores son NaN por ser demasiado pequeños para ser significativos.



Gráfica 6. Mapa de calor de correlación entre las variables.

Viendo la gráfica 6, podemos observar que surv2m y surv6m tienen alta correlación entre ellas mismas, a su vez podemos ver que charges, totcst y totmcst también poseen alta correlación por ser valores relacionados a los costos, dnrday y slos tienen alta correlación y adsl con adlsc también tienen alta correlación, totcst también tiene correlación con slos y dnrday, así que podemos concluir que estas se pueden reducir a charges, totcst y adsl, eliminando **dnrday, totmcst, slos y adlsc**.

Usando PCA se usará un criterio de 95% varianza, esto indica que se usarán los suficientes componentes hasta tener una varianza cercana a 95%, la cantidad de componentes obtenidos usando esa varianza fue de 38 componentes. 9 unidades por debajo de la cantidad de variables originales, lo que indica una reducción de 19.15%. Evaluando este PCA en los dos mejores modelos anteriormente obtenidos.

Modelo	MAE	RMSE	R2
Ensamble	0.064708	0.085958	0.875426
Paramétrico	0.038265	0.050130	0.957632

Tabla 4. Resultados obtenidos de usar PCA con los modelos anteriores

Como podemos ver en la tabla 4, entrenando los modelos que teníamos anteriormente con los nuevos componentes de PCA, obtenemos un decrecimiento en el rendimiento del modelo de ensamble pero una mejora en nuestro modelo Paramétrico, lo que podría indicar que hacer uso de PCA podría mejorar los resultados obtenidos muchísimo mas que lo obtenido aquí.

Para el UMAP se hará uso de un criterio de minimizar el número de componentes, mientras se intenta mantener la estructura, por lo cual usaremos 15 componentes en total, lo que indica una reducción del 79.45% sobre las variables originales. Después usamos ese UMAP para evaluar los mejores modelos anteriores y obtenemos.

Modelo	MAE	RMSE	R2
Ensamble	0.092696	0.124509	0.738633
Paramétrico	0.119938	0.154285	0.598673

Tabla 5. Resultados obtenidos de usar UMAP con los modelos anteriores

Como podemos ver en la tabla 5, entrenando nuestros modelos de ensamble y paramétrico, obtenemos unos valores de desempeño muchísimo peores comparados con los que se poseía anteriormente al uso de UMAP, a su vez se hizo pruebas con 5, 10 y 20 componentes, y todos dieron resultados similares, lo que podría indicar que la reducción dimensional mediante el uso de UMAP no es útil para este dataset.

El Random Forest Regressor o modelo ensamble mostró el mejor desempeño del estudio, con un R2 muy alto (0.969) y errores MAE y RMSE bajos, indicando predicciones precisas y consistentes. Al compararlo con otros modelos (SVC, regresión lineal, KNN y MLP), también sobresalió en todas las métricas. Estos resultados coinciden con trabajos previos, como el de Jared L. Katzman et al., que señalan al Random Survival Forest como uno de los métodos más efectivos para este tipo de problemas.

Dataset

[SUPPORT2 - UCI Machine Learning Repository](#)

Bibliografía

[Learning Patient-Specific Cancer Survival Distributions as a Sequence of Dependent Regressors](#)

[Personalized Survival Prediction with Contextual Explanation Networks](#)

[DeepSurv: personalized treatment recommender system using a Cox proportional hazards deep neural network](#)

[Let the Experts Speak: Improving Survival Prediction & Calibration via Mixture-of-Experts Heads](#)