Mejora de la estimación del esfuerzo en proyectos de software mediante métodos de sobremuestreo y aprendizaje computacional

Improving effort estimation in software projects using oversampling and machine learning methods

Beatriz Bedolla Martínez
Universidad Tecnológica de la Mixteca, Huajuapan de León, México
ie2014040408@gs.utm.mx
ORCID: 0009-0002-3474-3227

Raúl Cruz-Barbosa
Universidad Tecnológica de la Mixteca, Huajuapan de León, México
rcruz@gs.utm.mx
ORCID: 0000-0002-5494-7027

Iván Antonio García Pacheco
Universidad Tecnológica de la Mixteca, Huajuapan de León, México
ivan@mixteco.utm.mx
ORCID: 0000-0002-7594-6410

https://doi.org/10.36825/RITI.13.31.008

Recibido: Junio 22, 2025
Aceptado: Septiembre 30, 2025

Resumen: La predicción de la estimación del esfuerzo determina el tiempo que tomará desarrollar un software o los recursos que se requerirán para terminarlo en el tiempo establecido. Una alternativa actual para predecir las estimaciones es utilizar métodos de aprendizaje computacional, sin embargo, los conjuntos de datos disponibles públicamente generalmente contienen pocas muestras, por lo cual dichos métodos no pueden mejorar su efectividad. Entonces, es necesario aumentar el número de muestras mediante métodos de sobremuestreo. Por lo anterior, en este artículo se utilizan principalmente métodos de ensamble con combinaciones de sobremuestreo y submuestreo para analizar el efecto en el rendimiento de los regresores utilizados sobre conjuntos pequeños y medianos, evaluando así su efectividad en la mejora de la estimación del esfuerzo en proyectos de software, mediante el uso de medidas como MMRE, MAE, RMSE y Pred. Los resultados obtenidos de MMRE y Pred, principalmente, muestran que la aplicación de estas estrategias permite reducir los errores de predicción. Por tanto, la utilización de un modelo de ensamble adecuado, junto con las estrategias de sobremuestreo y submuestreo, permite mejorar la predicción del esfuerzo, especialmente en conjuntos de datos pequeños como COCOMO, Maxwell y Desharnais con alto desbalanceo en la distribución de sus muestras.

Palabras clave: *Estimación del Esfuerzo, Proyectos de Software, Aprendizaje Computacional, Predicción, Sobremuestreo, Regresión.*

© RITI. CC BY-NC

Abstract: Effort estimation prediction determines the time it will take to develop a software program or the resources required to complete it within the established timeframe. A current alternative for predicting estimates is to use machine learning methods. However, publicly available data sets generally contain few samples, so such methods cannot improve their effectiveness. Thus, it is necessary to increase the number of samples using oversampling methods. Therefore, this paper presents the use of ensemble methods with combinations of oversampling and undersampling to analyze the performance impact of the regressors used on small and medium-sized data sets. Moreover, their effectiveness in improving effort estimation in software projects using measures such as MMRE, MAE, RMSE, and Pred is also presented. The results obtained from MMRE and Pred, mainly show that the application of these strategies reduces prediction errors. Consequently, the use of an appropriate ensemble model, together with oversampling and undersampling strategies, improves effort prediction, especially on small data sets such as COCOMO, Maxwell, and Desharnais with highly unbalanced sample distributions.

Keywords: *Software Effort Estimation, Software Projects, Machine Learning, Prediction, Oversampling, Regression.*

1. Introducción

En la estimación de los proyectos de software, la estimación del esfuerzo determina la cantidad total de trabajo, generalmente medida en horas-persona o meses-persona, que se requiere para completar todas las tareas [1]. Sin embargo, a menudo los métodos tradicionales para realizar esta estimación no son efectivos al predecir el esfuerzo requerido por los proyectos de software, debido a la limitación de los datos, la complejidad, y la incapacidad de las organizaciones para adaptarse continuamente a los cambios.

En este sentido, estas limitaciones pueden superarse utilizando algoritmos de aprendizaje computacional, los cuales pueden identificar patrones y relaciones ocultas en los datos históricos para obtener predicciones más eficientes y precisas [2]. La eficacia de los métodos del aprendizaje computacional está estrechamente relacionada con la distribución del conjunto de datos empleado durante el entrenamiento. Por ejemplo, en conjuntos de datos desbalanceados, los métodos tienden a sesgarse hacia la clase mayoritaria; mientras que los ejemplos de clases minoritarias se predicen incorrectamente. De acuerdo con Belhaouari *et al.* [3], abordar este desbalance durante el entrenamiento del modelo es crucial para garantizar la mejor predicción posible.

Por otro lado, la Técnica de Sobremuestreo de Minorías Sintéticas (SMOTE, por sus siglas en inglés) genera muestras sintéticas de la clase minoritaria para problemas de clasificación [4]. Sin embargo, para problemas de regresión, donde los valores tienen una distribución continua en lugar de categorías, se han desarrollado diversas estrategias que buscan abordar el desbalanceo. Estas estrategias se enfocan en reducir el sesgo hacia los valores mayoritarios y mejorar la predicción en los casos de valores extremos [5]. Desde esta perspectiva, la investigación realizada por Sunda y Sinha [2] argumenta que se debe considerar un conjunto de datos que tenga suficientes muestras y, además, que proporcione información relevante para obtener una estimación satisfactoria [1].

Es decir, si se cuenta con conjuntos de datos con pocas muestras, además de exhibir poca representatividad del problema, se producirán resultados insatisfactorios en la estimación correspondiente. Por lo tanto, en el presente estudio se realiza un análisis de las configuraciones de la Técnica de Sobremuestreo Sintético de Minorías para Regresión con Ruido Gaussiano (SMOGN, por sus siglas en inglés), debido a que las investigaciones actuales no presentan las configuraciones utilizadas para SMOGN. Este análisis permitirá mostrar si los conjuntos aumentados de datos presentarán mayor representatividad de los proyectos de software involucrados y, en consecuencia, se obtendrán mejores resultados en la estimación del esfuerzo.

El resto del artículo está organizado de la siguiente manera. En la Sección 2 se presenta un análisis de los principales estudios relacionados con las técnicas de sobremuestreo utilizadas para problemas de regresión y para la estimación del esfuerzo. La Sección 3 describe el enfoque metodológico implementado para obtener la mejora de la estimación del esfuerzo en los conjuntos de datos seleccionado, los métodos utilizados, las medidas de evaluación y el preprocesamiento de los conjuntos utilizados. La Sección 4 muestra los resultados obtenidos aplicando las medidas de evaluación en cada uno de los experimentos descritos. Finalmente, la Sección 5 presenta una discusión sobre los resultados y hallazgos encontrados durante la evaluación de los métodos de sobremuestreo y resume las principales conclusiones del estudio.

© RITI. CC BY-NC

2. Estado del arte

Debido a diferentes situaciones del mundo real pueden surgir problemas de desbalanceo en los conjuntos de datos. La investigación de Avelino *et al.* [5], por ejemplo, estableció que, en el contexto de la clasificación en donde se utilizan variables numéricas independientes relacionadas con una variable dependiente categórica o de clase, el problema se resuelve sobremuestreando los ejemplos de la clase minoritaria; pero en tareas de regresión, donde la variable dependiente es continua, hallar una estrategia para el sobremuestreo es más difícil.

Considerando un proceso de regresión, cuando se predicen valores continuos o extremos, los modelos de aprendizaje computacional tienden a sesgarse hacia los valores más frecuentes, lo que afecta la predicción en los casos de mayor interés [6]. En este marco, SMOTEBoost, por ejemplo, es una técnica que combina al algoritmo SMOTE con métodos de boosting para generar variantes como AdaBoost.RT, AdaBoost+, AdaBoost.R2 y BEMBoost, con el fin de mejorar la predicción de los valores atípicos en problemas de regresión con datos desbalanceados. La técnica realiza iteraciones en las cuales se generan datos sintéticos en áreas donde los valores objetivos son escasos. Posteriormente, se realiza un ajuste de los pesos de cada muestra para que las prioritarias sean las que tengan un mayor peso en la predicción, de tal forma que, los casos donde el error de la predicción es mayor al umbral recibirán un peso mayor. De este modo, el modelo aprende a minimizar los errores al encontrar valores extremos que conduzcan finalmente a una mejor predicción [6].

Otra técnica empleada en problemas de regresión es SMOTER (i.e., SMOTE para regresión), la cual categoriza los valores continuos en valores extremos, normales y raros, generando datos sintéticos en regiones con baja densidad de datos utilizando medidas de distancia para considerar a los k-vecinos más cercanos, asegurando así que cada nueva muestra sea ubicada en regiones de baja densidad, sin alterar la distribución original. De esta manera se obtienen muestras en regiones donde los valores raros o extremos tienen menor representación [7].

En el contexto de la aplicación de estas técnicas a la estimación del esfuerzo en los proyectos de software, existe poca evidencia concluyente en la literatura especializada sobre su efectividad. La investigación de Jawa y Meena [8], por ejemplo, analizó el impacto de SMOTER en la estimación del esfuerzo en los proyectos de software. Para ello, se empleó la Regresión Lineal, Regresión Lasso, Regresión Ridge, Árbol de decisión y Bosque aleatorio a los conjuntos de datos de China [9], Maxwell [10] y COCOMO81 [11]. Además, se utilizaron como medidas de rendimiento el Error Relativo Medio, la Magnitud Media del Error Relativo y la precisión. Los principales hallazgos demostraron que SMOTER ayuda a reducir el error en los modelos empleados.

De igual forma, Durgesh *et al.* [1], propusieron un método de ensamble heterogéneo apilado que integra una máquina de aprendizaje extremo y una máquina de soporte vectorial para optimizar la estimación del esfuerzo en proyectos ágiles. Aunado a esto, se utilizó el método SMOTER para sobremuestrear el conjunto de datos de Desharnais [12]. Los resultados obtenidos mostraron que la predicción del esfuerzo mejora al utilizarse SMOTER, en comparación a cuando no se utiliza, reduciendo el error del modelo propuesto.

En la literatura existen investigaciones que utilizan SMOGN, para diversas tareas [13], [14], [15]. Por ejemplo, en [15], los autores utilizan diversas técnicas de sobremuestreo para regresión, incluyendo SMOGN, en la cual buscan predecir el número de días, que un paciente pediátrico puede estar sin ventilación mecánica en la unidad de cuidados intensivos, utilizando conjuntos pequeños con un mayor número de características. Aun cuando SMOGN ha sido utilizado para tareas de regresión con conjuntos de datos desbalanceados, no existen trabajos en la literatura que muestren el comportamiento del rendimiento utilizando diferentes configuraciones de sobremuestreo y submuestreo en la predicción de la estimación en conjuntos de datos de proyectos de software.

3. Materiales y métodos

La Figura 1 muestra la metodología utilizada para implementar el modelo de estimación del esfuerzo en los proyectos de software. Como se podrá observar, inicialmente se realizó una revisión de la bibliografía sobre las técnicas de sobremuestreo para regresión de los cuales se seleccionó SMOGN, debido a que puede generar un conjunto de muestras diversificado, a diferencia de SMOTER [15]. En virtud de lo anterior, se realizó la implementación de SMOGN para la estimación del esfuerzo. En última instancia, se llevó a cabo la evaluación de los métodos utilizados, cuyos resultados serán explicados más adelante con mayor detalle.

© RITI. CC BY-NC

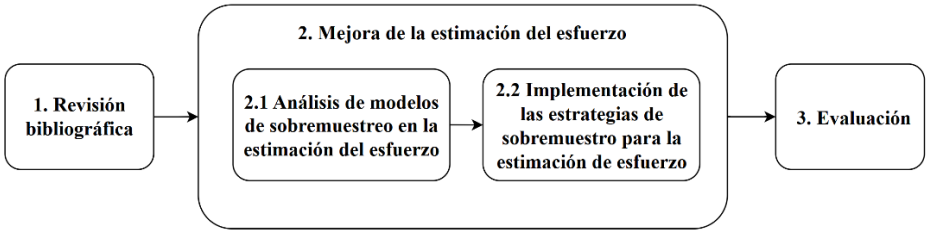

Figura 1. Metodología empleada.

La Figura 2 muestra el flujo de trabajo utilizado para el desarrollo de la mejora de la estimación. Por consiguiente, primero se realizó el preprocesamiento de los datos para verificar que los conjuntos de datos no tuvieran valores faltantes.

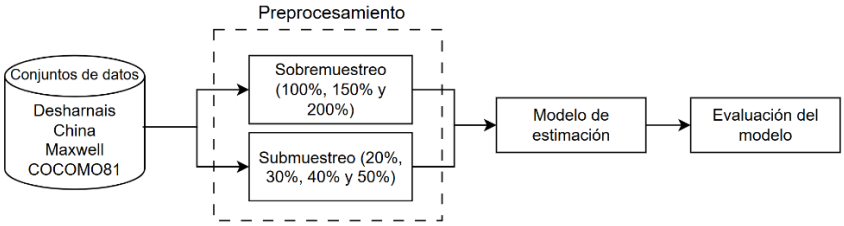

Figura 2. Flujo de trabajo para cada conjunto de datos.

Ulteriormente, se llevó a cabo la implementación del modelo de estimación. En consecuencia, se realizaron diversos experimentos para cada conjunto de datos, utilizando un porcentaje para sobremuestreo de 200%, 150% y 100%, y para submuestreo se utilizó 50%, 40%, 30% y 20% para cada conjunto de datos utilizado, por lo que se obtuvo un total de 12 combinaciones. Aunado a esto, se llevó a cabo la implementación del modelo de estimación para finalmente comparar si la estimación se mejoraba al considerar diferentes combinaciones de sobremuestreo y submuestreo. A continuación, las siguientes secciones describen las particularidades de este proceso en el contexto del estudio realizado.

3.1. Conjuntos de datos utilizados

Para este estudio se seleccionaron cuatro conjuntos de datos que han sido utilizados por investigadores en la obtención de la estimación del esfuerzo:

- China: El conjunto de datos de China [9] contiene 499 muestras y 19 atributos, de los cuales 18 son independientes y 1 dependiente que es el esfuerzo.
- Desharnais [12] consta de 81 muestras (proyectos de software) y 12 atributos, de los cuales 11 son independientes (ID del proyecto, Experiencia del equipo, Experiencia del gerente, Fin de año, Duración, Transacciones, Entidades, Puntos no ajustados, Factor de ajuste, Punto ajustados y Lenguaje) y un atributo dependiente que es el esfuerzo.
- COCOMO81: Este conjunto de datos es público, consta de 63 muestras (proyectos) y 17 atributos, 16 atributos independientes y el esfuerzo como atributo dependiente [11]. El esfuerzo está definido por persona-mes.
- Maxwell: Este conjunto de datos es público, consta de 62 muestras (proyectos) y 27 atributos, 16 atributos independientes y el esfuerzo como atributo dependiente. El esfuerzo también está definido por persona-mes [10].

3.2. Métodos utilizados

Esta sección describe los métodos utilizados para la mejora de la estimación del esfuerzo. Antes de abordarlos, es importante describir SMOGN, la cual es una extensión de SMOTER que integra el ruido Gaussiano para la

generación de las muestras. SMOGN utiliza la interpolación entre vecinos cercanos, por lo que permite equilibrar los conjuntos de datos al mejorar la representación de valores atípicos.

Como resultado de lo anterior, se reduce la tendencia a que el modelo se sesgue hacia los valores más frecuentes sin modificar la distribución original. El ruido Gaussiano tiene como propósito simular la variabilidad natural de los datos, permitiendo así que el modelo aprenda a manejar la incertidumbre y, por consiguiente, realice predicciones más robustas [16].

En la práctica es necesario que el porcentaje de sobremuestreo para un conjunto de datos no sea menor al 100%, ya que al hacerlo se producirían pocas muestras para los pequeños conjuntos de datos. Además, tampoco es posible incrementar el porcentaje del submuestreo más allá de un 50%, ya que esto excedería la eliminación de datos importantes dentro del conjunto de datos y se perdería información importante, además de que podría afectar negativamente el rendimiento del modelo a entrenar.

Una vez planteadas las limitaciones para el sobremuestreo y submuestreo, se procede a describir cada uno de los modelos utilizados.

- El Regresor de Soporte Vectorial (SVR, por sus siglas en inglés), por ejemplo, intenta predecir un valor continuo, buscando una función que se desvié lo menor posible de los valores reales, permitiendo tener un pequeño margen (i.e., épsilon) alrededor de las predicciones [17].
- Una Máquina de Aprendizaje Extremo (ELM, por sus siglas en inglés) es una red neuronal de una sola capa oculta, que aprende de los patrones de los datos y realiza predicciones asignando pesos aleatoriamente al calcular el peso de salida de cada predicción [1].
- El árbol de decisión (DT, por sus siglas en inglés), es un modelo predictivo utilizado en problemas de clasificación y regresión que divide los datos en ramas basándose en condiciones o reglas sobre las características, hasta obtener una predicción en una hoja [18].
- El bosque aleatorio (RF, por sus siglas en inglés) es un conjunto de árboles de decisión que combinan sus predicciones para obtener los resultados más robustos y precisos [17].
- La Regresión Lineal (LR, por sus siglas en inglés) es un método estadístico que se utiliza para modelar la relación entre dos variables, una o varias independientes y otra dependiente. Tiene como objetivo estimar los valores de los coeficientes de la ecuación lineal prediciendo los valores de la variable dependiente utilizando las variables independientes [17].
- La Regresión Lasso (LASSO, por sus siglas en inglés) es una variante de la regresión lineal que incorpora una regularización tipo L1, la cual penaliza la suma de los valores absolutos de los coeficientes del modelo [17].
- La Regresión Ridge (RR, por sus siglas en inglés) es una variante de la regresión lineal que incorpora una penalización tipo L2, que consiste en penalizar las predicciones al utilizar el error cuadrático, esto ayuda a reducir el sobreajuste [8].

Finalmente, los modelos de ensamble son técnicas que combinan múltiples modelos, los cuales pueden categorizarse como:

- Bagging, que entrena en paralelo los modelos, utilizando particiones del conjunto de datos proporcionalmente.
- Stacking o apilamiento, que combina modelos de forma que integra las predicciones de todos como entrada a un metamodelo.
- Boosting, que entrena modelos de forma secuencial, concentrándose en corregir los errores del anterior modelo [19].

Para formar distintos ensambles, se utilizaron los modelos individuales anteriormente descritos, los cuales fueron integrados en un método diferente de ensamble heterogéneo que toma las predicciones de ambos métodos y realiza, por medio de una regla de combinación, las predicciones finales. Estas predicciones pueden ser por la media, mediana o la mediana ponderada por el inverso del rango. Para la experimentación se empleó la regla de combinación de la mediana ponderada por el inverso del rango, debido a que proporcionó mejores resultados para cada configuración. La Tabla 1, señala los modelos integrados en cada ensamble.

© RITI. CC BY-NC

Tabla 1. Ensambles utilizados.

Ensamble	Modelos involucrados
Ensamble 1 (E1)	ELM + SVR, Metamodelo RF
Ensamble 2 (E2)	ELM + SVR + DT + RF + LR, Metamodelo SVR
Ensamble 3 (E3)	ELM + SVR + DT + RF + LR, Metamodelo RF
Ensamble 4 (E4)	ELM + SVR + LR, Metamodelo RF
Ensamble 5 (E5)	RF + RF + RF, Metamodelo RF
Ensamble 6 (E6)	ELM + SVR + LASSO, Metamodelo RF
Ensamble 7 (E7)	ELM + SVR + Ridge, Metamodelo RF

Fuente: Elaboración propia.

3.2.1. Medidas de evaluación

Para establecer y comparar que técnica de sobremuestreo mejora las predicciones de la estimación del esfuerzo se utilizan distintas de medidas de evaluación. Los modelos de estimación empleados fueron evaluados utilizando las siguientes medidas de evaluación. El Error Absoluto Medio (MAE, por sus siglas en inglés) es la suma del promedio de todos los errores absolutos de la diferencia del esfuerzo real con el esfuerzo estimado [20].

$$MAE = \frac{1}{N}\sum_{i=1}^{N}|EsfuerzoReal_i - EsfuerzoEstimado_i| \quad (1)$$

en donde $EsfuerzoReal_i$ es el esfuerzo real del proyecto y el $EsfuerzoEstimado_i$ es el esfuerzo que se obtuvo de la predicción del modelo de estimación, y *n* representa al número total de proyectos. Por otra parte, la raíz del Error Cuadrático Medio (RMSE, por sus siglas en inglés) es la raíz cuadrada de la diferencia cuadrada media entre el esfuerzo real y el esfuerzo estimado [17].

$$RMSE = \sqrt{\frac{(EsfuerzoReal_i - EsfuerzoEstimado_i)^2}{n}} \quad (2)$$

La Magnitud Media del Error Relativo (MMRE, por sus siglas en inglés) es el promedio de cada observación (i.e., proyecto) *i* del Error Relativo Medio (MRE por sus siglas en inglés), que está representado en la siguiente ecuación [21].

$$MRE_i = \frac{|EsfuerzoReal_i - EsfuerzoEstimado_i|}{EsfuerzoReal_i}$$

$$MMRE = \frac{1}{N}\sum_{i=1}^{N} MRE_i \quad (3)$$

Pred(l) representa qué porcentaje de lo obtenido en MRE es menor o igual a l.

$$Pred(l) = \frac{k}{n} \quad (4)$$

Donde:

n es el total de proyectos y

k es el total de proyectos que tienen un MRE menor o igual a l [22].

3.2.2. Preprocesamiento del conjunto de datos

De acuerdo con Rahman *et al.* [23], el preprocesamiento de un conjunto de datos es relevante para mejorar el rendimiento del modelo de estimación. Para mostrar la necesidad de recurrir a la técnica de sobremuestreo se utilizó el conjunto de datos Desharnais descrita en la sección 3, mediante una gráfica de la variable dependiente (esfuerzo). Como se puede observar en la Figura 3, la distribución de la característica dependiente (i.e., esfuerzo) se concentra en un rango de 0 a 15000 horas-persona, por lo que existe un desbalance significativo de los datos.

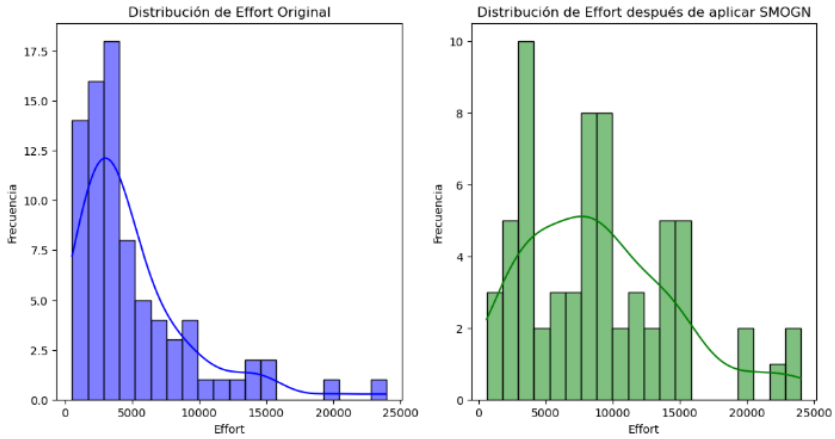

Figura 3. Distribución de Desharnais.

Por lo tanto, es necesario balancear el conjunto de datos para permitir una mejor variabilidad y, como consecuencia, facilitar a los modelos de estimación la predicción adecuada de ésta. Para hacer esto con SMOGN, es importante mencionar que se utiliza el parámetro samp_method = 'extreme' para permitir que se realice un sobremuestreo adecuado, ya que de no hacerlo se corre el riesgo de reducir las muestras del conjunto de datos.

Una vez que se aplica SMOGN, se obtiene una mejor distribución de los datos, lo cual se muestra al lado derecho de la Figura 3. Esto evidencía la importancia del sobremuestreo al evitar sesgos y diversificar las muestras. Por último, el tema de particionamiento de los conjuntos de datos es importante. Por lo tanto, se consideró que, debido a que los conjuntos de datos de Desharnais, Maxwell y COCOMO constan de pocas muestras, se utilizara un particionamiento de 80% para entrenamiento y 20% para pruebas. Para el conjunto de datos de China se consideró una partición del 70% para entrenamiento y 30% para pruebas.

4. Resultados

La experimentación realizada en esta sección tiene los siguientes objetivos: (i) analizar el efecto del sobremuestreo y submuestreo en el rendimiento de los regresores utilizados y (ii) la influencia de estos en conjuntos de datos pequeños y medianos. A continuación, se presentan los resultados obtenidos para cada uno de estos objetivos:

4.1. Análisis del efecto del sobremuestreo y submuestreo en el rendimiento de los regresores

En relación con lo anterior, se evaluaron diversos modelos de regresión aplicando porcentajes de sobremuestreo del 100%, 150% y 200%, así como de submuestreo del 50%, 40%, 30% y 20%. Para cada configuración se realizó una búsqueda gruesa de hiperparámetros utilizando la técnica de *Grid Search*, que posteriormente se mejoró utilizando una búsqueda fina alrededor de los encontrados. En la Tabla 2 se indican los intervalos de búsqueda de cada modelo utilizado en el *Grid Search*.

Después de terminar la búsqueda fina, para cada modelo de ensamble se utilizaron los valores de cada parámetro encontrado en dicha búsqueda, los cuales se muestran en la Tabla 3.

© RITI. CC BY-NC

Tabla 2. Intervalos de búsqueda de hiperparámetros.

Modelo	Parámetros y valores
ELM	n_hidden: [50, 100, 150, 200, 300] activation: ['relu', 'tanh', 'sigmoid']
SVR	C: [1, 10, 100] Kernel: ['rbf', 'linear'] Gamma: ['scale', 'auto', 0.01, 0.001] Épsilon: [0.01, 0.1, 1]
DT	max_depth: None = D, [D + 5, D + 10, D + 20] i.e. sí D = 6 → [6, 11, 16, 26] min_samples_split: [1, 2, 3, 4, 5]
RF	n_estimators: [10, 50, 100, 150, 200] max_depth: None = D, [D + 5, D + 10, D + 20] min_samples_split: [2, 5, 10]
Lasso	alpha: [0.0001, 0.001, 0.01, 0.1, 1, 10] max_iter: [10, 50, 100, 200]
Ridge	alpha: [0.0001, 0.001, 0.01, 0.1, 1, 10] max_iter: [10, 50, 100, 200]

Fuente: Elaboración propia.

Tabla 3. Hiperparámetros encontrados con la búsqueda fina.

Modelo	Maxwell	COCOMO	Desharnais	China
ELM	n_hidden: 100 activation: 'relu'	n_hidden: 150 activation: 'tanh'	n_hidden: 100 activation: 'sigmoid'	n_hidden: 50 activation: 'relu'
SVR	C: 0.5 Kernel: 'linear' Gamma: 'scale'	C: 100 Kernel: 'linear' Gamma: 'scale'	C: 20 Kernel: 'linear' Gamma: 'scale'	C: 1 Kernel: 'linear' Gamma: 'scale'
DT	max_depth: 6 min_samples_split: 2	max_depth: 11 min_samples_split: 3	max_depth: 8 min_samples_split: 2	max_depth: 11 min_samples_split: 3
RF	n_estimators: 250 max_depth: 9 min_samples_split: 3	n_estimators: 187 max_depth: 9 min_samples_split: 11	n_estimators: 250 max_depth: 8 min_samples_split: 2	n_estimators: 75 max_depth: 21 min_samples_split: 3
Lasso	alpha: 15 max_iter: 202	alpha: 15 max_iter: 52	alpha: 15 max_iter: 48	alpha: 15
Ridge	alpha: 15 max_iter: 8	alpha: 15 max_iter: 8	alpha: 15 max_iter: 48	alpha: 0.15

Fuente: Elaboración propia.

En las Tablas 4 a 11, los resultados en negritas representan el error mínimo o máximo Pred obtenido de cada combinación, mientras que los números subrayados indican los valores inferiores al mínimo error de las referencias [1] y [8] presentados en la columna 2, donde los valores con guion indican que el autor no presentó resultados de dichos modelos para efectos de comparación, permitiendo identificar la mejor combinación para cada conjunto de datos respectivamente.

Además, se muestran los valores obtenidos para las combinaciones de los porcentajes de sobremuestreo y submuestreo. Por ejemplo, en la columna 3 de las tablas se muestran los valores obtenidos al utilizar un sobremuestreo del 100% y un porcentaje del 50% en submuestreo, y en la columna 10 se indican los valores obtenidos al utilizar un porcentaje de submuestreo del 20% y un porcentaje de sobremuestreo del 150%. El submuestreo se acotó a 50%, ya que eliminar más de la mitad de las muestras de un conjunto pequeño afectaría al sobremuestreo correspondiente. Para el sobremuestreo se limitó el aumento de datos a 200% por cuestiones de tiempo computacional y espacio del artículo.

La Tabla 4 ilustra los valores de la medida de evaluación MMRE. El modelo de ensamble 4 muestra un mínimo MMRE con un submuestreo del 30% consistente con todos los porcentajes de sobremuestreo del conjunto de datos Maxwell.

© RITI. CC BY-NC

Tabla 4. Resultados de MMRE sobre el conjunto Maxwell.

Modelo	[8]	100				150				200			
		50	**40**	**30**	**20**	**50**	**40**	**30**	**20**	**50**	**40**	**30**	**20**
ELM	-	1.117	3.409	3.051	3.027	1.117	3.409	3.051	3.027	2.008	2.454	3.573	5.786
SVR	-	1.086	**0.510**	1.393	3.369	1.086	**0.510**	1.393	3.369	0.968	2.660	1.646	**0.312**
E1	-	0.950	3.015	0.325	0.852	0.950	3.015	0.325	0.852	1.266	2.239	**0.376**	0.748
DT	0.60	2.491	1.276	0.373	1.539	2.491	1.276	0.373	1.539	**0.667**	1.306	0.520	1.539
RF	**0.49**	1.203	1.118	1.476	2.289	1.203	1.118	1.476	2.289	0.972	0.702	2.086	2.071
LR	0.86	5.357	3.111	6.905	7.029	5.357	3.111	6.905	7.029	6.531	3.935	5.902	7.029
E2	-	1.076	2.596	3.165	3.369	1.076	2.596	3.165	5.027	1.012	2.078	4.599	3.332
E3	-	1.169	1.930	0.385	0.880	1.169	1.930	0.385	0.880	0.935	1.809	0.510	0.739
E4	-	0.934	2.909	**0.304**	**0.849**	0.934	2.909	**0.304**	**0.849**	1.197	2.379	0.460	0.778
E5	-	1.019	0.541	0.832	0.960	1.019	0.541	0.832	0.960	0.845	**0.573**	0.686	1.748
Lasso	0.85	4.061	5.233	3.876	4.983	4.061	5.233	3.876	4.983	3.710	4.766	7.435	15.827
Ridge	0.8	1.463	1.027	1.689	0.883	1.463	1.027	1.689	0.883	1.690	0.677	2.567	1.067
E6	-	0.935	3.567	0.429	0.851	0.935	3.567	0.429	0.851	1.385	2.658	0.395	0.793
E7	-	**0.910**	2.914	0.382	0.862	**0.910**	2.914	0.382	0.862	1.251	2.122	0.426	0.813

Fuente: Elaboración propia.

Tabla 5. Resultados de PRED sobre el conjunto Maxwell.

Modelo	[8]	100				150				200			
		50	**40**	**30**	**20**	**50**	**40**	**30**	**20**	**50**	**40**	**30**	**20**
ELM	-	14.286	20	0	0	14.286	20	0	0	14.286	33.333	0	33.333
SVR	-	28.571	20	25	33.333	28.571	20	25	33.333	28.571	16.667	0	33.333
E1	-	0	**60**	50	33.333	0	**60**	50	33.333	14.286	16.667	**50**	33.333
DT	56.6	0	0	50	0	0	0	50	0	42.857	16.667	25	0
RF	**66.6**	14.286	20	25	0	14.286	20	25	0	42.857	16.667	0	33.333
LR	33.3	14.286	0	0	0	14.286	0	0	0	28.571	33.333	0	0
E2	-	**42.857**	20	25	33.333	**42.857**	20	25	0	14.286	16.667	0	33.333
E3	-	14.286	40	50	33.333	14.286	40	50	33.333	14.286	33.333	25	33.333
E4	-	28.571	20	**75**	33.333	28.571	20	**75**	33.333	14.286	16.667	25	33.333
E5	-	28.571	0	0	**66.667**	28.571	0	0	**66.667**	**57.143**	16.667	25	33.333
Lasso	33.3	14.286	20	0	33.333	14.286	20	0	33.333	14.286	33.333	0	33.333
Ridge	33.3	28.571	20	50	**66.667**	28.571	20	50	**66.667**	28.571	**50**	0	**66.667**
E6	-	28.571	20	**75**	33.333	28.571	20	**75**	33.333	28.571	0	**50**	33.333
E7	-	14.286	**60**	**75**	33.333	14.286	**60**	**75**	33.333	14.286	16.667	25	33.333

Fuente: Elaboración propia.

La Tabla 5, muestra los resultados de la medida Pred para el conjunto de datos de Maxwell. Esto evidencía que los modelos de ensamble proporcionan un mejor rendimiento. De manera similar a la Tabla 4, los mejores resultados se obtuvieron con un submuestreo de 30% de manera consistente con los sobremuestreos, a excepción del correspondiente al 200%.

La Tabla 6 presenta los resultados de MMRE obtenidos del conjunto de datos COCOMO. En ella se aprecia que los modelos de ensamble obtienen mejores resultados, a diferencia de los modelos lineales. También, se confirma que un mayor porcentaje de sobremuestreo tiende a mejorar el rendimiento.

Los resultados de la medida Pred para el conjunto de datos COCOMO refuerzan la importancia que tiene la selección tanto del submuestreo como del sobremuestreo para mejorar los resultados de cada medida, como se muestra en la Tabla 7.

La Tabla 8 muestra los resultados de la medida MAE sobre el conjunto de datos Desharnais, la cual enfatiza que el sobremuestreo del 200% y un submuestreo del 20% muestra mejores resultados. El sobremuestreo permite que los modelos tengan más información de los proyectos y realicen una mejor predicción.

Estos hallazgos se respaldan con los valores de RMSE expuestos en la Tabla 9, donde dicha combinación proporciona los mejores resultados en la mayoría de los modelos. Al mismo tiempo, se muestra que con mayor porcentaje de sobremuestreo, combinado con submuestreo mínimo, se obtienen mejores resultados que [1] en la mayoría de los modelos.

© RITI. CC BY-NC

Tabla 6. Resultados de MMRE sobre el conjunto COCOMO.

Modelo	[8]	100				150				200			
		50	40	30	20	50	40	30	20	50	40	30	20
ELM	-	8.552	27.602	9.477	36.294	8.552	38.715	9.477	36.294	13.128	18.716	14.336	36.726
SVR	-	**1.966**	1.436	2.095	1.730	**1.966**	1.554	2.095	1.730	1.754	0.960	4.034	2.058
E1	-	4.927	0.410	1.345	**0.419**	4.927	0.396	1.345	**0.419**	**0.967**	3.891	1.872	0.543
DT	**1.50**	6.113	1.481	3.357	0.463	6.113	1.400	3.357	0.463	3.569	2.133	3.021	0.463
RF	2.60	4.846	6.782	10.771	32.725	4.564	6.584	10.771	32.725	3.121	2.918	5.041	17.356
LR	15.10	27.233	57.093	90.606	67.817	27.233	77.136	90.606	67.817	42.744	90.294	80.006	67.817
E2	-	2.964	1.941	1.230	7.315	2.983	3.374	1.230	7.315	5.938	4.080	**0.570**	3.711
E3	-	4.835	0.801	1.437	1.350	4.875	0.730	1.437	1.350	1.817	11.917	1.597	1.375
E4	-	5.109	0.385	1.424	1.446	5.154	0.379	1.424	1.446	2.251	17.686	1.299	1.772
E5	-	2.777	2.805	5.591	125.88	2.777	1.580	50.591	125.88	2.116	**0.629**	1.674	**0.408**
Lasso	14.40	21.517	80.576	21.574	40.612	17.398	69.614	21.574	40.612	24.456	39.698	20.134	41.615
Ridge	7.90	3.765	5.186	5.272	3.466	3.913	4.621	5.272	3.466	3.546	3.203	23.459	41.533
E6	-	4.625	**0.382**	**0.896**	1.619	4.351	**0.368**	**0.896**	1.619	1.921	4.633	1.534	0.685
E7	-	4.851	0.456	1.591	0.588	4.830	0.474	1.591	0.588	1.131	3.169	1.566	0.561

Fuente: Elaboración propia.

Tabla 7. Resultados de PRED sobre el conjunto COCOMO.

Modelo	[8]	100				150				200			
		50	40	30	20	50	40	30	20	50	40	30	20
ELM	-	14.286	33.333	**50**	0	14.286	33.333	**50**	0	14.286	33.333	0	0
SVR	-	0	0	0	0	0	0	0	0	0	16.667	**20**	0
E1	-	14.286	50	0	0	14.286	50	0	0	**28.571**	33.333	0	0
DT	5.20	14.286	0	0	**33.333**	14.286	0	0	**33.333**	14.286	33.333	**20**	**33.333**
RF	5.20	0	16.667	0	0	14.286	16.667	0	0	0	16.667	0	0
LR	10.50	0	0	0	0	0	0	0	0	14.286	0	0	0
E2	-	14.286	16.667	0	0	14.286	16.667	0	0	14.286	16.667	**20**	0
E3	-	0	16.667	0	**33.333**	0	33.333	0	**33.333**	14.286	33.333	0	0
E4	-	14.286	50	0	0	14.286	50	0	0	14.286	16.667	0	0
E5	-	0	0	25	**33.333**	0	16.667	25	**33.333**	14.286	0	0	**33.333**
Lasso	**15.70**	14.286	0	0	0	0	16.667	0	0	0	0	0	0
Ridge	10.50	14.286	33.333	**25**	0	14.286	33.333	**25**	0	14.286	0	0	0
E6	-	14.286	50	**25**	0	14.286	**66.667**	**25**	0	0	16.667	0	0
E7	-	**28.571**	**66.667**	**25**	0	**28.571**	**66.667**	**25**	0	14.286	**50**	0	0

Fuente: Elaboración propia.

Tabla 8. Resultados de MAE sobre el conjunto Desharnais.

Modelo	[1]	100				150				200			
		50	40	30	20	50	40	30	20	50	40	30	20
ELM	2197.23	2635.464	3317.129	**1163.781**	2236.072	1124.566	2544.932	1712.223	2635.464	1930.620	1730.650	1541.007	681.105
SVR	2240.83	1932.708	4698.005	2876.802	1993.897	1780.477	1373.977	2428.250	1932.708	2208.839	2919.554	1707.245	1226.556
E1	**1412.25**	1559.469	3880.499	1327.387	2357.653	1253.635	2411.281	1941.788	1559.469	1812.393	2097.874	1977.610	**511.518**
DT	-	1688.889	2816.125	1628.667	1223.250	1371.300	**588.000**	3940.417	1688.889	1201.200	3170.562	2420.000	522.200
RF	-	**1147.556**	2320.337	1616.916	2413.325	1174.379	855.971	2399.905	**1147.556**	958.168	1413.642	2042.532	1851.489
LR	-	1801.268	5003.879	2783.909	1297.459	1626.519	2727.312	1743.861	1801.268	1494.612	2434.117	**1357.163**	802.310
E2	-	1817.067	**1975.910**	1984.278	1014.827	1109.806	750.742	**1556.744**	1817.067	**746.593**	**1035.729**	1362.444	2456.164
E3	-	1396.378	2127.857	1321.542	1898.046	**998.256**	1105.553	2475.058	1396.378	1185.152	1104.673	1998.111	618.122
E4	-	1543.348	3897.054	1560.417	2250.298	1199.841	2390.392	2014.465	1543.348	1704.162	2138.788	1943.981	581.000
E5	-	1436.589	2131.000	2048.278	2279.298	1409.132	1312.102	1764.911	1436.589	1237.787	1315.465	1781.288	1337.386
Lasso	-	1539.271	3700.676	2690.793	**896.438**	1310.808	2424.557	1692.225	1539.271	1513.932	2422.742	1654.956	705.682
Ridge	-	1452.588	5105.293	2880.351	1266.470	1542.292	2689.911	1562.467	1452.588	1666.188	2629.497	1358.806	994.420
E6	-	1510.533	3610.232	1577.333	1949.839	1214.453	2397.665	1969.910	1510.533	1706.927	2181.992	1942.477	576.464
E7	-	1476.512	3916.982	1466.597	2272.056	1201.546	2414.874	1935.307	1476.512	1729.809	2143.252	1964.136	561.781

Fuente: Elaboración propia.

© RITI. CC BY-NC

Tabla 9. Resultados de RMSE sobre el conjunto Desharnais.

Modelo	[1]	100				150				200			
		50	40	30	20	50	40	30	20	50	40	30	20
ELM	2846.64	3749.178	4106.164	**1776.638**	2886.252	1747.227	3767.687	2231.986	3749.178	2951.384	2171.618	2332.914	1075.651
SVR	3104.94	2613.889	6215.155	5468.670	3176.622	2866.704	2137.975	3402.516	2613.889	3257.932	3637.812	2835.936	2276.922
E1	**2360.51**	2487.138	4555.597	1872.047	2843.996	1887.943	3144.335	2425.891	2487.138	2579.646	2500.325	2708.159	**552.118**
DT	-	2234.670	4027.112	2291.595	2153.035	1821.796	**876.168**	4557.225	2234.670	1926.578	5181.461	3680.258	764.027
RF	-	**1758.043**	3038.255	2086.285	2582.859	1806.617	1193.891	2541.857	**1758.043**	1746.124	2101.480	2280.807	1954.596
LR	-	2162.458	6711.815	4229.866	1733.133	1847.930	3501.960	2314.658	2162.458	2065.256	3018.417	2004.753	1007.971
E2	-	2543.687	2731.890	2632.282	1362.099	1664.622	1092.424	**1855.407**	2543.687	**1646.940**	1691.375	**1790.951**	3118.431
E3	-	2081.205	2784.843	1807.579	2527.973	**1564.396**	1406.025	2870.148	2081.205	1911.715	**1605.262**	2702.532	680.496
E4	-	2390.023	4863.757	2114.144	2748.120	1843.343	3018.605	2402.198	2390.023	2397.850	2490.104	2602.652	620.768
E5	-	1846.759	**2582.376**	2594.789	2750.923	2100.001	1420.908	2388.627	1846.759	1879.233	1946.159	2173.271	1749.702
Lasso	-	1901.349	4557.139	3497.851	**1213.073**	1703.147	3189.451	2120.623	1901.349	2078.641	3007.605	2385.797	1017.381
Ridge	-	1820.292	6849.999	5002.692	2367.580	1902.439	3481.000	2233.743	1820.292	2148.145	3198.876	2179.083	1388.292
E6	-	2406.862	4392.982	2139.815	2271.333	1846.279	3019.035	2343.553	2406.862	2404.327	2550.375	2576.631	617.636
E7	-	2373.880	4878.113	1993.932	2793.784	1835.609	3045.705	2449.931	2373.880	2423.660	2489.664	2638.556	581.744

Fuente: Elaboración propia.

La Tabla 10 muestra los resultados obtenidos para la medida MMRE sobre el conjunto de datos China, en la cual se muestra que la selección del modelo combinado con la adecuada estrategia de combinación del sobremuestreo y submuestreo es vital para la obtención de estimaciones precisas, ya que, en este caso, la mayoría de los modelos de ensamble no presentan resultados competitivos.

Los resultados de *Pred* sobre el conjunto de datos de China, presentados en la Tabla 11, muestran que las combinaciones de sobremuestreo y submuestreo sí influyen en la evaluación de cada modelo, destacando que RF y el Ensamble 3 alcanzaron los mejores resultados.

Tabla 10. Resultados de MMRE sobre el conjunto China.

Modelo	[8]	100				150				200			
		50	40	30	20	50	40	30	20	50	40	30	20
ELM	-	1.083	1.209	0.475	1.481	1.316	0.782	0.702	1.254	0.677	1.309	0.680	1.617
SVR	-	0.105	0.113	**0.098**	**0.104**	**0.064**	**0.062**	**0.097**	**0.130**	**0.067**	**0.079**	**0.101**	**0.150**
E1	-	0.111	0.422	0.289	0.528	0.100	0.459	0.469	0.526	0.094	0.710	0.419	1.000
DT	0.09	0.133	0.126	0.155	0.540	0.108	0.188	0.125	0.474	0.121	0.142	0.136	0.440
RF	**0.06**	0.099	**0.108**	**0.098**	0.505	0.088	0.117	0.099	0.473	0.092	0.138	0.123	0.328
LR	0.12	0.427	0.368	0.337	0.773	0.388	0.357	0.296	1.041	0.356	0.327	0.362	0.735
E2	-	1.618	1.168	1.981	5.220	2.004	1.660	1.443	5.139	1.686	1.207	2.454	4.463
E3	-	**0.086**	0.225	0.166	0.449	0.082	0.221	0.203	0.428	0.073	0.305	0.218	0.697
E4	-	0.116	0.394	0.258	0.535	0.093	0.438	0.439	0.501	0.085	0.643	0.395	0.924
E5	-	0.127	0.121	0.142	0.438	0.109	0.157	0.126	0.467	0.105	0.153	0.130	0.264
Lasso	0.16	0.526	0.608	0.513	1.111	0.451	0.576	0.635	1.421	0.383	0.610	0.765	1.375
Ridge	0.12	0.428	0.368	0.337	0.773	0.390	0.359	0.310	1.075	0.359	0.327	0.382	0.735
E6	-	0.110	0.403	0.277	0.544	0.093	0.444	0.433	0.492	0.086	0.658	0.103	0.935
E7	-	0.118	0.394	0.260	0.535	0.095	0.438	0.441	0.497	0.087	0.643	0.396	0.924

Fuente: Elaboración propia.

Tabla 11. Resultados de PRED sobre el conjunto China.

Modelo	[8]	100				150				200			
		50	40	30	20	50	40	30	20	50	40	30	20
ELM	-	26.316	27.869	54.348	32.258	40.260	38.710	48.936	46.875	44.156	30.159	50.000	42.424
SVR	-	92.105	91.803	**93.478**	**93.548**	97.403	**96.774**	93.617	**87.500**	97.403	**98.413**	**93.750**	**81.818**
E1	-	92.105	39.344	76.087	51.613	97.403	56.452	55.319	59.375	93.506	44.444	62.500	51.515
DT	94.5	88.158	88.525	89.130	70.968	92.208	74.194	85.106	71.875	92.208	90.476	85.417	72.727
RF	**98.6**	**96.053**	**98.361**	**93.478**	83.871	**98.701**	91.935	**97.872**	81.250	97.403	92.063	**93.750**	78.788
LR	87.6	61.842	60.656	63.043	51.613	63.636	66.129	57.447	56.250	71.429	71.429	64.583	63.636
E2	-	22.368	13.115	17.391	16.129	16.883	16.129	17.021	21.875	15.584	15.873	18.750	18.182

© RITI. CC BY-NC

E3	-	**96.053**	72.131	86.957	83.871	96.104	74.194	85.106	84.375	**98.701**	66.667	81.250	69.697
E4	-	88.158	42.623	76.087	51.613	93.506	56.452	55.447	62.500	96.104	46.032	66.667	60.606
E5	-	88.158	90.164	84.783	74.194	89.610	85.484	82.979	78.125	94.805	87.302	83.333	**81.818**
Lasso	83.5	57.895	55.738	52.174	58.065	57.143	50.000	48.936	46.875	61.039	49.206	50.000	51.515
Ridge	87.2	63.158	60.656	63.043	51.613	62.338	66.129	53.191	56.250	72.727	71.429	62.500	63.636
E6	-	90.789	40.984	71.739	58.065	96.104	56.452	57.447	62.500	97.403	46.032	66.667	57.576
E7	-	88.158	42.623	73.913	51.613	92.208	56.452	57.447	62.500	96.104	46.032	66.667	60.606

Fuente: Elaboración propia.

4.2. Influencia del sobremuestreo en conjuntos pequeños y medianos

Por otro lado, la Tabla 12 muestra la comparativa de los mejores resultados reportados en la literatura y los obtenidos en el presente estudio. De acuerdo con las tablas (Tabla 4 a la Tabla 11) de la sección anterior, y resumidas en la Tabla 12, se observa de manera general que el sobremuestreo impacta de manera positiva en el rendimiento de las medidas de evaluación para todos los conjuntos de datos analizados.

Tabla 12. Comparativa de medidas por cada conjunto de datos.

Maxwell			
Modelo	MMRE	Modelo	Pred
RF [8]	0.490	RF [8]	66.6
Ensamble 4	**0.304**	Ensamble 4	**75.0**
COCOMO			
Modelo	MMRE	Modelo	Pred
DT [8]	1.500	DT [8]	5.2
Ensamble 6	**0.368**	Ensamble 6	**66.6**
Desharnais			
Modelo	MAE	Modelo	RMSE
ELM + SVR [1]	1412.25	ELM + SVR [1]	2360.51
Ensamble 1	**511.51**	Ensamble 1	**552.11**
China			
Modelo	MMRE	Modelo	Pred
RF [8]	0.06	RF [8]	98.6
SVR	**0.06**	RF	**98.7**

Fuente: Elaboración propia.

De manera específica, también se observa que los conjuntos pequeños (i.e., Maxwell, COCOMO y Desharnais) fueron mayormente beneficiados en el incremento de su rendimiento, lo que indica que el aumento de sus muestras favorece en el rendimiento de los modelos utilizados. En el caso del conjunto de datos China, el incremento en rendimiento fue pequeño, lo cual era esperado ya que este conjunto contiene un mayor número de muestras de proyectos.

5. Conclusión

La estimación del esfuerzo en los proyectos es crucial para determinar el tiempo necesario para desarrollar el software. Obtener la predicción de la estimación utilizando aprendizaje computacional es adecuado, ya que permite determinar a tiempo el esfuerzo requerido, mejorando así la planificación del proyecto.

En este contexto, en este estudio se presentaron los resultados de evaluar el impacto del uso de técnicas de sobremuestreo, en conjuntos de datos desbalanceados con pocas muestras mediante la aplicación de SMOGN, en conjunto con modelos de regresión aplicados a cuatro conjuntos de datos, con el objetivo de mejorar la estimación del esfuerzo. La implementación de modelos individuales como SVR, DT, RF, Regresión Lasso, Regresión Ridge y modelos de ensamble combinando algunos de estos, permitió visualizar y analizar las diferencias relevantes en el rendimiento de cada uno de los modelos, evaluándolos mediante las medidas MMRE, Pred, MAE y RMSE, utilizadas comúnmente en la literatura.

Los resultados obtenidos mostraron una reducción de los valores de las medidas mencionadas, como consecuencia de las estrategias de combinaciones de sobremuestreo y submuestreo empleadas. Por lo tanto, se concluye que la utilización de un modelo de ensamble adecuado, junto con las estrategias de combinación, puede

© RITI. CC BY-NC

mejorar la precisión de la predicción del esfuerzo, especialmente en conjuntos de datos pequeños y con alto desbalanceo en la distribución de sus muestras.

6. Agradecimientos

B. Bedolla agradece el apoyo que la Secretaría de Ciencia, Humanidades, Tecnología e Innovación (SECIHTI) ha otorgado para la realización de este estudio a través de su convocatoria de becas nacionales para estudios de posgrado.

7. Referencias

[1] Durgesh, D. V. S., Saket, M. V. S., Reddy, B. R. (2023). Improving software effort estimation with heterogeneous stacked ensemble using SMOTER over ELM and SVR base learners. En R. Morusupalli, T. S. Dandibhotla, V. V. Atluri, D. Windridge, P. Lingras, V. R. Komati (Eds.), *Multi-disciplinary trends in artificial intelligence* (pp. 442–448). Springer Nature Switzerland. https://doi.org/10.1007/978-3-031-36402-0_41

[2] Sunda, N., Sinha, R. R. (2023). *Optimizing effort estimation in agile software development: Traditional vs. advanced ML methods.* IEEE International Conference on Communication, Security and Artificial Intelligence (ICCSAI). Greater Noida, India. https://doi.org/10.1109/ICCSAI59793.2023.10421235

[3] Belhaouari, S. B., Islam, A., Kassoul, K., Al-Fuqaha, A., Bouzerdoum, A. (2024). Oversampling techniques for imbalanced data in regression. *Expert Systems with Applications, 252*, 1-19. https://doi.org/10.1016/j.eswa.2024.124118

[4] Chawla, N. V., Bowyer, K. W., Hall, L. O., Kegelmeyer, W. P. (2002). SMOTE: Synthetic minority over-sampling technique. *Journal of Artificial Intelligence Research, 16*, 321–357. https://doi.org/10.1613/jair.953

[5] Avelino, J. G., Cavalcanti, G. D. C., Cruz, R. M. O. (2024). Resampling strategies for imbalanced regression: A survey and empirical analysis. *Artificial Intelligence Review, 57*, 82–124. https://doi.org/10.1007/s10462-024-10724-3

[6] Moniz, N., Ribeiro, R., Cerqueira, V., Chawla, N. (2018*). SMOTEBoost for regression: Improving the prediction of extreme values*. IEEE 5th International Conference on Data Science and Advanced Analytics (DSAA). Turin, Italy. https://doi.org/10.1109/DSAA.2018.00025

[7] Torgo, L., Ribeiro, R. P., Pfahringer, B., Branco, P. (2013). SMOTE for regression. En L. Correia, L. P. Reis, J. Cascalho (Eds.), *Progress in Artificial Intelligence* (pp. 378–389). Springer. https://doi.org/10.1007/978-3-642-40669-0_33

[8] Jawa, M., Meena, S. (2022). *Software effort estimation using synthetic minority over-sampling technique for regression (SMOTER)*. IEEE 3rd International Conference for Emerging Technology (INCET). Belgaum, India. https://doi.org/10.1109/INCET54531.2022.9824043

[9] Yun, F. H. (2025). *China: Effort estimation dataset*. Zenodo. https://zenodo.org/records/268446

[10] Li, Y. (2025). *Effort estimation: Maxwell*. Zenodo. https://zenodo.org/records/268461

[11] Kaggle. (2025). *Effort-estimation-on-cocomo-dataset*. https://kaggle.com/code/vanlocbk1996/effort-estimation-on-cocomo-dataset

[12] Esteves, A. (2025). *Software effort estimation*. https://github.com/yy2111/Software-Effort-Estimation/blob/master/Datasets/02.desharnais.csv

[13] Bhattacharyya, A., Srijith, K., Behera, R. P., Dasgupta, A., Chakraborty, R. S. (2024). *A study on effects of synthetic data for predicting the remaining useful life of aluminium electrolytic capacitors using bagging-based ensemble learning*. International Conference on Advances in Data-driven Computing and Intelligent Systems (ADCIS). Goa, India. https://doi.org/10.1007/978-981-99-9518-9_40

[14] Qi, L., Zhihao, L., Jianxiao, Z. I. (2024). A SMOGN-based MPSO-BP model to predict the height of a hydraulically conductive fracture zone. *Coal Geology & Exploration, 52* (11), 72–85. https://cge.researchcommons.org/journal/vol52/iss11/7/

[15] Rad, M., Rafiei, A., Grunwell, J., Kamaleswaran, R. (2025). Tackling the small imbalanced horizontal dataset regressions by stability selection and SMOGN: A case study of ventilation-free days prediction in

© RITI. CC BY-NC

the pediatric intensive care unit and the importance of PRISM. *International Journal of Medical Informatics, 196*. https://doi.org/10.1016/j.ijmedinf.2025.105809

[16] Branco, P., Torgo, L., Ribeiro, R. P. (2017). *SMOGN: A pre-processing approach for imbalanced regression.* First International Workshop on Learning with Imbalanced Domains: Theory and Applications (LIDTA). Skopje, Macedonia. https://proceedings.mlr.press/v74/branco17a/branco17a.pdf

[17] Rahman, M., Sarwar, H., Kader, M. D. A., Gonçalves, T., Tin, T. T. (2024). Review and empirical analysis of machine learning-based software effort estimation. *IEEE Access, 12*, 85661–85680. https://doi.org/10.1109/ACCESS.2024.3404879

[18] Abid, M., Bukhari, S., Saqlain, M. (2025). Enhancing software effort estimation in healthcare informatics: A comparative analysis of machine learning models with correlation-based feature selection. *Sustainable Machine Intelligence, 10*, 50–66. https://doi.org/10.61356/SMIJ.2025.10451

[19] Mienye, I. D., Sun, Y. (2022). A survey of ensemble learning: Concepts, algorithms, applications, and prospects. *IEEE Access, 10*, 99129–99149. https://doi.org/10.1109/ACCESS.2022.3207287

[20] Varshini, A. G. P., Kumari, K. A., Janani, D., Soundariya, S. (2021). Comparative analysis of machine learning and deep learning algorithms for software effort estimation. *Journal of Physics: Conference Series, 1767*, 1-11. https://doi.org/10.1088/1742-6596/1767/1/012019

[21] Şengüneş, B., Öztürk, N. (2023). An artificial neural network model for project effort estimation. *Systems, 11* (2), 1-22. https://doi.org/10.3390/systems11020091

[22] Zakrani, A., Hain, M., Idri, A. (2019). Improving software development effort estimating using support vector regression and feature selection. *IAES International Journal of Artificial Intelligence, 8* (4), 399–410. https://doi.org/10.11591/ijai.v8.i4.pp399-410

[23] Rahman, M., Roy, P. P., Ali, M., Goncalves, T., Sarwar, H. (2023). Software effort estimation using machine learning technique. *International Journal of Advanced Computer Science and Applications, 14* (4), 822–827. https://doi.org/10.14569/IJACSA.2023.0140491

© RITI. CC BY-NC