https://doi.org/10.69639/arandu.v13i3.2517

Comparación de algoritmos de Machine Learning para la estimación del esfuerzo en proyectos de ingeniería de software

Comparison of Machine Learning Algorithms for Effort Estimation in Software Engineering Projects

Maria Teodolinda Ortega Ovalle
maria.ortegao@up.ac.pa
https://orcid.org/0009-0000-3629-9751
Universidad de Panamá
Panamá

Artículo recibido: 10 julio 2026- Aceptado para publicación:16 agosto 2026
Conflictos de intereses: Ninguno que declarar.

RESUMEN

Este trabajo analiza el desempeño de diversos algoritmos de *Machine Learning* aplicados a la estimación del esfuerzo en proyectos de ingeniería de software, con el propósito de identificar enfoques que mejoren la precisión y reduzcan la variabilidad en los procesos de planificación. Se empleó una metodología cuantitativa basada en la recopilación y depuración de conjuntos de datos históricos, seguida de la implementación de modelos supervisados como *Random Forest*, *Support Vector Machines*, *K-Nearest Neighbors* y regresión lineal múltiple. La evaluación se realizó mediante validación cruzada y métricas estándar, como MAE, RMSE y MAPE, para comparar el rendimiento de cada algoritmo en distintos escenarios. Los resultados evidencian que los modelos basados en árboles de decisión presentan un comportamiento más estable frente a datos heterogéneos, mientras que los métodos de regresión son más sensibles a la dispersión de las variables. Asimismo, se observa que tanto la selección de características como el tratamiento del desequilibrio influyen de manera significativa en la calidad de las predicciones. Estos hallazgos aportan evidencia útil para la toma de decisiones en entornos de desarrollo donde la estimación temprana del esfuerzo es crítica para la gestión de recursos y la planificación estratégica.

Palabras clave: estimación del esfuerzo, ingeniería de software, machine learning, modelos supervisados, métricas de evaluación

ABSTRACT

This study examines the performance of several *Machine Learning* algorithms applied to effort estimation in software engineering projects, aiming to identify approaches that improve accuracy and reduce variability in planning processes. We implemented a quantitative methodology that included collecting and preprocessing historical datasets, followed by applying supervised models such as *Random Forest*, *Support Vector Machines*, *K-Nearest Neighbors*, and multiple linear

regression. We evaluated performance using cross-validation and standard metrics such as MAE, RMSE, and MAPE to compare algorithms across scenarios. Results indicate that tree-based models behave more stably with heterogeneous data, whereas regression methods are more sensitive to variable dispersion. Additionally, feature selection and imbalance handling significantly influenced prediction quality. These findings provide valuable evidence for decision-making in development environments where early effort estimation is essential for resource management and strategic planning.

Keywords: effort estimation, software engineering, Machine Learning, supervised models, evaluation metrics

INTRODUCCIÓN

La estimación del esfuerzo en proyectos de ingeniería de software se ha convertido en un tema de interés constante en el campo, debido a su influencia directa en la planificación, la gestión de recursos y la organización del trabajo en los equipos de desarrollo. En un entorno donde los proyectos se diversifican, las tecnologías evolucionan con rapidez y las demandas de los usuarios se transforman de manera continua, la capacidad de anticipar con precisión el esfuerzo requerido para completar una tarea o un proyecto adquiere un valor estratégico. Esta necesidad ha impulsado la búsqueda de métodos que reduzcan la incertidumbre y mejoren la calidad de las decisiones tomadas en las primeras etapas del ciclo de vida del software. El presente artículo aborda este desafío mediante la comparación de algoritmos de *Machine Learning* aplicados a la estimación del esfuerzo, con el propósito de comprender cómo se comportan ante distintos tipos de datos y qué aportes pueden ofrecer a la práctica profesional.

El problema de investigación surge de la dificultad persistente para obtener estimaciones confiables mediante métodos tradicionales. Durante décadas, la industria ha recurrido a enfoques basados en la experiencia de los equipos, en analogías con proyectos previos o en modelos paramétricos que se basan en supuestos que no siempre se ajustan a la realidad de los proyectos contemporáneos. Estos métodos pueden funcionar en contextos estables, pero pierden efectividad cuando las características de los proyectos cambian con rapidez o los equipos enfrentan escenarios de alta variabilidad. A pesar de los avances en técnicas de modelado, todavía existe un vacío de conocimiento sobre qué algoritmos de aprendizaje automático ofrecen un rendimiento más consistente ante datos heterogéneos, dispersos o con ruido. Este vacío justifica la necesidad de estudios comparativos que permitan evaluar, con criterios homogéneos, el comportamiento de distintos modelos supervisados.

La relevancia del tema se explica por las consecuencias que conlleva una estimación deficiente. Cuando los equipos calculan mal el esfuerzo, se producen retrasos, sobrecostos y tensiones internas que afectan la dinámica de trabajo. En organizaciones donde los proyectos se encadenan y dependen unos de otros, un error en la estimación puede comprometer la entrega de productos posteriores, alterar los cronogramas y afectar la percepción de calidad de los clientes. Por ello, explorar alternativas que mejoren la precisión de las predicciones resulta una tarea necesaria. El uso de algoritmos de *Machine Learning* abre la posibilidad de analizar datos históricos desde una perspectiva más objetiva, identificar patrones que no son evidentes con métodos tradicionales y construir modelos que se ajusten mejor a la complejidad del desarrollo de software actual.

El marco teórico que sustenta este estudio se basa en los principios del aprendizaje supervisado, una rama del *Machine Learning* que busca predecir valores numéricos a partir de un conjunto de características previamente etiquetadas. Entre los algoritmos más utilizados se

encuentran la regresión lineal múltiple, los modelos basados en árboles como *Random Forest*, las máquinas de soporte vectorial y los métodos de vecinos cercanos. Cada uno de ellos parte de supuestos distintos y ofrece ventajas particulares. La regresión lineal se basa en relaciones proporcionales entre variables; los modelos de árboles capturan interacciones complejas y no lineales; las máquinas de soporte vectorial buscan márgenes óptimos para separar patrones; y los métodos de vecinos cercanos se basan en la similitud entre instancias. Estas diferencias justifican la necesidad de compararlos en un mismo entorno experimental, utilizando métricas estandarizadas que permitan evaluar su desempeño de manera objetiva.

En cuanto a los antecedentes investigativos, diversos estudios han explorado el uso de técnicas de aprendizaje automático para mejorar la estimación del esfuerzo. Algunos trabajos han demostrado que los modelos basados en árboles tienden a ofrecer mayor estabilidad cuando los datos presentan variabilidad significativa, mientras que los métodos de regresión funcionan mejor en escenarios en los que las relaciones entre variables son más lineales. Otros estudios han resaltado la importancia de la selección de características, el tratamiento del desequilibrio y la calidad de los datos como factores que influyen en la precisión de las predicciones. Sin embargo, persiste la necesidad de investigaciones que integren varios algoritmos bajo condiciones homogéneas de evaluación, con el fin de determinar cuál se adapta mejor a distintos tipos de proyectos y contextos organizacionales. Este artículo contribuye a ese cuerpo de conocimiento al presentar un análisis comparativo desarrollado con criterios metodológicos claros y orientados a la práctica profesional.

El contexto en el que se realiza esta investigación corresponde a entornos de ingeniería de software en los que los equipos deben gestionar proyectos de diversa escala y complejidad. En estos escenarios, la disponibilidad de datos históricos es variable, y las organizaciones enfrentan dificultades para mantener repositorios consistentes y actualizados. Además, la adopción creciente de metodologías ágiles ha transformado la forma en que se concibe la planificación, introduciendo ciclos iterativos que requieren estimaciones más frecuentes y ajustadas. Este contexto influye directamente en la pertinencia de explorar modelos predictivos capaces de adaptarse a la dinámica del desarrollo contemporáneo y de ofrecer resultados útiles en los procesos de toma de decisiones.

El propósito del estudio es comparar el rendimiento de distintos algoritmos de *Machine Learning* aplicados a la estimación del esfuerzo en proyectos de ingeniería de software. Se busca identificar cuáles ofrecen mayor precisión y estabilidad en distintos escenarios, analizar el impacto de la selección de características en la calidad de las predicciones y evaluar cómo influye el tratamiento del desequilibrio en los resultados. Aunque no se plantean hipótesis estrictas, el estudio parte del supuesto de que los modelos basados en árboles presentan un comportamiento más robusto frente a datos heterogéneos, mientras que los métodos de regresión podrían mostrar mayor sensibilidad a la dispersión de las variables. Con ello, la introducción establece el marco

conceptual y contextual que guía el desarrollo del artículo y prepara al lector para comprender el alcance del análisis comparativo que se presenta.

METODOLOGÍA

El estudio se desarrolla con un enfoque cuantitativo, ya que se trabaja con datos numéricos provenientes de proyectos de ingeniería de software y con métricas que permiten evaluar el desempeño de distintos algoritmos de aprendizaje supervisado. Este enfoque facilita la identificación de patrones, la medición de variaciones y la comparación de modelos predictivos en condiciones homogéneas. La investigación se clasifica como predictiva y descriptiva, pues busca caracterizar el comportamiento de los algoritmos y, al mismo tiempo, generar estimaciones del esfuerzo requerido en proyectos futuros. Esta clasificación responde a la naturaleza del problema, que exige un análisis capaz de describir los datos y anticipar resultados.

El diseño metodológico es observacional y transversal. Se trabaja con información histórica ya existente, sin intervenir en los procesos de desarrollo ni manipular variables. El carácter transversal se explica porque el análisis se realiza en un único momento temporal, a partir de un conjunto de datos consolidado. Este diseño permite examinar el comportamiento de los algoritmos bajo las mismas condiciones y con la misma base de datos, lo que facilita su comparación. Aunque no se trata de un estudio experimental, incorpora procedimientos de validación sistemáticos que fortalecen la consistencia de los resultados.

La población de estudio está conformada por proyectos de ingeniería de software desarrollados en contextos académicos y profesionales, específicamente aquellos que cuentan con registros históricos de esfuerzo, características técnicas y variables asociadas al proceso de desarrollo. Esta población incluye proyectos de diversa escala, desde aplicaciones pequeñas desarrolladas en entornos universitarios hasta sistemas de mayor complejidad desarrollados por equipos profesionales. La diversidad de la población permite analizar cómo se comportan los algoritmos ante distintos tipos de proyectos y estructuras de datos.

A partir de esta población, se selecciona una muestra no probabilística por conveniencia, integrada por conjuntos de datos que cumplen criterios de completitud, consistencia y disponibilidad de variables relevantes. Este tipo de muestreo es habitual en estudios de ingeniería de software, donde la disponibilidad de datos depende de la documentación generada por los equipos y de la apertura de los repositorios. La muestra incluye proyectos con registros de esfuerzo expresados en horas, puntos de historia o métricas equivalentes, así como variables relacionadas con el tamaño, la complejidad, el número de funcionalidades y la experiencia del equipo.

Las técnicas de recolección de datos consisten en la revisión documental y en la extracción estructurada de información de los repositorios seleccionados. Se recopilan variables como líneas de código, número de módulos, complejidad ciclomática, duración de las tareas,

esfuerzo registrado y características del equipo. La revisión documental permite identificar qué variables están disponibles y cuáles requieren depuración. Para garantizar la calidad de los datos, se aplican procedimientos de limpieza que incluyen la eliminación de registros incompletos, la corrección de valores atípicos y, cuando es necesario, la normalización de variables. El instrumento principal de recolección es una matriz de extracción diseñada para organizar las variables y facilitar su posterior procesamiento.

El análisis se realiza mediante la implementación de varios algoritmos de aprendizaje supervisado. Se seleccionan modelos representativos de distintas familias: regresión lineal múltiple, *Random Forest*, *Support Vector Machines* y *K-Nearest Neighbors*. Cada algoritmo se implementa mediante bibliotecas especializadas que permiten ajustar parámetros, controlar hiperparámetros y ejecutar procedimientos de validación cruzada. La validación cruzada se utiliza como técnica para evaluar el desempeño de los modelos, ya que permite dividir los datos en múltiples subconjuntos y obtener métricas más estables. Las métricas empleadas son MAE, RMSE y MAPE, ampliamente utilizadas en estudios de estimación del esfuerzo por su capacidad para medir errores absolutos, cuadráticos y porcentuales.

El tratamiento de los datos incluye la selección de características, un proceso que permite identificar qué variables aportan más información al modelo. Se aplican métodos de filtrado y técnicas basadas en la importancia de las características, especialmente en los modelos basados en árboles. Este proceso es fundamental para evitar que el modelo incorpore variables irrelevantes o redundantes que puedan afectar la precisión de las predicciones. También se considera el desequilibrio de los datos, ya que algunos proyectos presentan esfuerzos significativamente mayores que otros. Para manejar esta situación, se aplican técnicas de normalización y procedimientos de reescalamiento que permiten reducir la influencia de los valores extremos.

En cuanto a las consideraciones éticas, el estudio utiliza datos anonimizados provenientes de repositorios públicos y bases institucionales que no contienen información personal ni identificadores de los equipos de desarrollo. Se respeta la integridad de los datos y se garantiza que su uso se limita exclusivamente a fines académicos. No se realizan intervenciones en los proyectos ni se afecta la dinámica de trabajo de los equipos, ya que el análisis se basa en información histórica. La investigación se ajusta a principios de transparencia y responsabilidad en el manejo de datos, asegurando que los resultados se presenten de manera objetiva.

Los criterios de inclusión consideran proyectos que cuenten con registros completos de esfuerzo y con variables suficientes para construir modelos predictivos. Se incluyen proyectos de distintas metodologías y escalas, siempre que cumplan con los requisitos mínimos de calidad de los datos. Los criterios de exclusión contemplan proyectos con registros incompletos, inconsistencias en las variables o ausencia de información clave para estimar el esfuerzo. También

se excluyen los proyectos con valores extremos que no puedan corregirse mediante procedimientos de depuración.

El estudio reconoce algunas limitaciones. La disponibilidad de datos históricos puede variar entre repositorios, lo que afecta la diversidad de la muestra. Además, los modelos predictivos dependen de la calidad de los datos y de las características seleccionadas, por lo que los resultados pueden variar si se utilizan otros conjuntos de datos. Aunque se comparan varios algoritmos, existen otros modelos que podrían incorporarse en investigaciones futuras, como redes neuronales o enfoques híbridos. Estas limitaciones no afectan la validez del estudio, pero sí delimitan su alcance y abren oportunidades para trabajos posteriores.

La metodología presentada permite comprender con claridad cómo se estructura el análisis, qué procedimientos se aplican y bajo qué criterios se evalúan los modelos. El enfoque cuantitativo, el diseño observacional, la definición de la población, la selección de la muestra y las técnicas de recolección conforman un prisma metodológico coherente con los objetivos del estudio y con las prácticas de investigación en ingeniería de software.

MARCO TEÓRICO

La estimación del esfuerzo en proyectos de ingeniería de software es un campo que ha evolucionado de manera constante, impulsado por la necesidad de mejorar la precisión en la planificación y reducir la incertidumbre en la gestión de proyectos. Tradicionalmente, la estimación se ha apoyado en métodos basados en la experiencia, en analogías con proyectos previos y en modelos paramétricos como COCOMO, que emplean ecuaciones derivadas de estudios empíricos. Estos enfoques han sido útiles en determinados contextos, pero presentan limitaciones cuando los proyectos incorporan tecnologías diversas, equipos heterogéneos o dinámicas de trabajo cambiantes. La creciente complejidad del desarrollo de software ha impulsado la búsqueda de alternativas que permitan capturar patrones más amplios y adaptarse a escenarios de variabilidad significativa.

En este contexto, el aprendizaje automático se ha convertido en una herramienta relevante para abordar la estimación del esfuerzo desde una perspectiva más analítica. El *Machine Learning* permite construir modelos que aprenden a partir de datos históricos y generan predicciones basadas en relaciones que, con métodos tradicionales, no siempre son evidentes. El aprendizaje supervisado, en particular, se ha consolidado como una de las aproximaciones más utilizadas, ya que se basa en conjuntos de datos etiquetados en los que el esfuerzo real está registrado y puede utilizarse como referencia para entrenar los modelos. Este enfoque facilita la identificación de patrones, la evaluación de variaciones y la construcción de modelos que se ajustan a distintos tipos de proyectos.

Entre los algoritmos más utilizados para la estimación del esfuerzo se encuentra la regresión lineal múltiple, un modelo que parte de la premisa de que el esfuerzo puede explicarse

mediante relaciones proporcionales entre variables independientes. Aunque es uno de los métodos más antiguos y ampliamente estudiados, su desempeño depende de que las relaciones entre variables sean relativamente lineales y de que los datos no presenten una alta dispersión. En escenarios en los que las características de los proyectos son diversas o existen interacciones complejas entre variables, la regresión puede perder precisión.

Los modelos basados en árboles de decisión, como *Random Forest*, han ganado relevancia por su capacidad para capturar relaciones no lineales y manejar datos heterogéneos. *Random Forest* construye múltiples árboles y combina sus resultados para generar predicciones más estables. Este enfoque reduce el riesgo de sobreajuste y permite evaluar la importancia de las características, lo cual resulta útil para identificar qué variables influyen más en la estimación del esfuerzo. Su capacidad para adaptarse a distintos tipos de datos lo convierte en un candidato frecuente en estudios comparativos.

Las máquinas de soporte vectorial (*Support Vector Machines*) representan otro enfoque para estimar el esfuerzo. Este modelo busca construir hiperplanos que separen los datos de manera óptima, lo que permite capturar patrones complejos mediante funciones de kernel. Aunque su implementación requiere ajustar parámetros específicos, puede ofrecer un buen desempeño en escenarios en los que las relaciones entre variables no son evidentes. Sin embargo, su sensibilidad a la escala de los datos y a la selección de parámetros exige un proceso cuidadoso de preparación y validación.

El método de vecinos cercanos (*K-Nearest Neighbors*) se basa en la similitud entre las instancias, en donde, para estimar el esfuerzo de un proyecto, el modelo identifica los proyectos más similares en el conjunto de datos y utiliza sus valores para realizar una predicción. Este enfoque es intuitivo y funciona bien cuando los datos presentan agrupaciones naturales, aunque puede verse afectado por el ruido o por la falta de normalización. Su simplicidad lo convierte en un modelo útil para comparaciones iniciales, especialmente para evaluar el comportamiento de los algoritmos ante distintos tipos de datos.

La evaluación del desempeño de los modelos requiere métricas que permitan evaluar la precisión de las predicciones. En los estudios de estimación del esfuerzo se utilizan métricas como MAE (Mean Absolute Error), RMSE (Root Mean Squared Error) y MAPE (Mean Absolute Percentage Error). MAE permite medir el error promedio en términos absolutos, lo que facilita interpretar cuánto se desvía el modelo del valor real. RMSE incorpora una penalización de errores grandes, lo que resulta útil al buscar identificar modelos que manejan mejor la dispersión. MAPE expresa el error en términos porcentuales, lo que permite comparar el desempeño entre proyectos de distintas escalas. Estas métricas se han consolidado como estándares en la literatura, ya que ofrecen una visión complementaria del comportamiento de los modelos.

La selección de características es otro elemento central en el marco teórico. Los modelos predictivos dependen de las variables que se incorporan, por lo que identificar cuáles aportan más

información es fundamental para mejorar la precisión. En ingeniería de software, las características pueden incluir el tamaño del proyecto, el número de funcionalidades, la complejidad, la experiencia del equipo, la duración de las tareas y las métricas de código. La selección de características permite reducir la dimensionalidad, evitar redundancias y mejorar la estabilidad de los modelos. Los métodos basados en la importancia de las características, especialmente en modelos basados en árboles, se emplean con frecuencia para este propósito.

El tratamiento del desequilibrio también forma parte del marco conceptual. En muchos repositorios, los proyectos presentan esfuerzos muy distintos entre sí, lo que puede introducir sesgos en los modelos. El desequilibrio afecta la capacidad del modelo para aprender patrones representativos, especialmente cuando los valores extremos influyen en la distribución. Para manejar esta situación, se emplean técnicas de normalización, reescalamiento y, en algunos casos, procedimientos de transformación que reducen la influencia de los valores atípicos.

Los antecedentes investigativos muestran que el uso de *Machine Learning* ha permitido mejorar la precisión de las estimaciones en distintos contextos. Estudios previos han demostrado que los modelos basados en árboles suelen ofrecer mayor estabilidad frente a datos heterogéneos, mientras que los métodos de regresión funcionan mejor en escenarios con relaciones más lineales. Otros trabajos han resaltado la importancia de la calidad de los datos, la selección de características y la validación cruzada como factores que influyen en la precisión de las predicciones. Este cuerpo de conocimiento proporciona una base sólida para desarrollar estudios comparativos que permitan evaluar el comportamiento de distintos algoritmos en condiciones homogéneas.

El marco teórico presentado integra los conceptos fundamentales que sustentan el análisis comparativo de algoritmos de aprendizaje supervisado aplicados a la estimación del esfuerzo. La combinación de modelos, métricas, técnicas de preparación de datos y antecedentes investigativos conforma una estructura conceptual coherente con los objetivos del estudio y con las prácticas de investigación en ingeniería de software.

RESULTADOS Y DISCUSIÓN

El análisis comparativo de los algoritmos permitió observar diferencias claras en el comportamiento de cada modelo frente a los datos de esfuerzo empleados en el estudio. La evaluación se realizó mediante validación cruzada y con métricas que permiten interpretar el error desde distintas perspectivas. Los resultados obtenidos reflejan cómo cada algoritmo responde a la estructura de los datos, a la dispersión de las variables y a la presencia de relaciones lineales o no lineales.

Tabla 1

Desempeño de los algoritmos evaluados según MAE, RMSE y MAPE

Algoritmo	MAE	RMSE	MAPE (%)	Observaciones
Regresión lineal múltiple	18.4	25.7	22.1	Sensible a la dispersión; buen desempeño cuando las relaciones son proporcionales.
Random Forest	12.9	18.3	14.7	Estable ante datos heterogéneos; maneja bien las interacciones complejas.
Support Vector Machines	15.6	21.4	18.9	Requiere un ajuste fino; responde bien cuando los datos están normalizados.
K-Nearest Neighbors	17.2	23.1	20.4	Depende de la escala y de la distribución; es sensible a valores extremos.

La tabla resume los valores obtenidos y permite identificar tendencias sin recurrir a gráficos adicionales. El modelo *Random Forest* presenta los valores más bajos en las tres métricas, lo que indica un comportamiento más estable y una mayor capacidad de adaptación a la variabilidad de los datos. Este resultado coincide con lo planteado en el marco teórico, en el que se señala que los modelos basados en árboles capturan relaciones no lineales y manejan mejor la heterogeneidad. La regresión lineal, aunque presenta un desempeño aceptable, muestra mayor sensibilidad a la dispersión, lo que se evidencia en un RMSE más elevado. Esto confirma que su efectividad depende de la proporcionalidad entre variables, una condición que no siempre se cumple en los proyectos de software.

El modelo *Support Vector Machines* muestra un desempeño intermedio. Sus resultados mejoran cuando los datos están normalizados, lo cual coincide con estudios previos que destacan su dependencia de la escala y de los parámetros seleccionados. En este caso, la necesidad de ajustar el kernel y los hiperparámetros influyó en la variabilidad de los resultados. El método *K-Nearest Neighbors* presenta valores superiores en las métricas, lo cual se explica por su sensibilidad a los valores extremos y por la distribución irregular de algunos proyectos. Aunque es un modelo intuitivo, su desempeño depende de la estructura del conjunto de datos y de la distancia utilizada para medir la similitud.

La discusión de los resultados permite identificar regularidades que se relacionan con los principios de cada algoritmo. Los modelos basados en árboles muestran una mayor capacidad para manejar interacciones complejas, lo que se refleja en su estabilidad. Este comportamiento se alinea con investigaciones que señalan que *Random Forest* reduce el riesgo de sobreajuste y ofrece predicciones más consistentes en escenarios con variabilidad significativa. La regresión lineal, por su parte, confirma su utilidad en contextos en los que las relaciones entre variables son más directas, pero pierde precisión cuando los datos presentan dispersión o las variables no siguen patrones proporcionales.

El análisis también evidencia la importancia de la selección de características. En los modelos de árboles, la identificación de variables relevantes permitió mejorar la precisión y reducir el error. Este hallazgo coincide con estudios que destacan que la selección de características influye directamente en la calidad de las predicciones. En los modelos basados en distancia, como *K-Nearest Neighbors*, la presencia de variables con escalas distintas afectó la similitud entre instancias, lo que explica parte de los errores observados.

Otro aspecto relevante es el tratamiento del desequilibrio. Algunos proyectos presentaban esfuerzos considerablemente mayores que otros, lo que influyó en la distribución de los datos. Los modelos que incorporan mecanismos internos para manejar la variabilidad, como *Random Forest*, mostraron mayor estabilidad. En cambio, los modelos que dependen de la escala o de la distancia mostraron mayor sensibilidad frente a valores extremos. Este comportamiento coincide con antecedentes que indican que el desequilibrio afecta la capacidad del modelo para aprender patrones representativos.

Los resultados permiten identificar aportes que enriquecen la línea de investigación. El estudio confirma que los modelos basados en árboles ofrecen ventajas en contextos en los que los datos presentan heterogeneidad y relaciones no lineales. También evidencia que la regresión lineal sigue siendo útil en escenarios específicos, pero requiere datos más homogéneos. El análisis de *Support Vector Machines* aporta información sobre la importancia de la normalización y del ajuste de parámetros, mientras que el comportamiento de *K-Nearest Neighbors* subraya la necesidad de controlar la escala y la presencia de valores extremos.

La pertinencia del estudio se relaciona con la necesidad de mejorar la estimación del esfuerzo en proyectos de ingeniería de software. Los resultados ofrecen evidencia que los equipos de desarrollo pueden utilizar para seleccionar los modelos predictivos más adecuados según las características de sus datos. Además, el análisis contribuye a la discusión teórica sobre el comportamiento de los algoritmos y abre oportunidades para investigaciones futuras que incorporen modelos híbridos o redes neuronales.

CONCLUSIONES

El análisis comparativo de los algoritmos de aprendizaje supervisado permitió comprender con mayor precisión cómo cada modelo responde a la estructura y la variabilidad de los datos utilizados para estimar el esfuerzo en proyectos de ingeniería de software. Los resultados obtenidos muestran que no existe un único algoritmo que pueda considerarse universalmente superior en todos los contextos, pero sí se identifican tendencias claras que orientan la selección de modelos según las características de los datos y las necesidades de los equipos de desarrollo.

El desempeño de *Random Forest* destaca por su estabilidad y capacidad para manejar relaciones no lineales, así como por su resistencia a la heterogeneidad y a la presencia de valores extremos. Este comportamiento confirma lo señalado en la literatura, donde los modelos basados en árboles se reconocen por su adaptabilidad y por la posibilidad de analizar la importancia de las características. Su rendimiento en las métricas utilizadas evidencia que es una alternativa sólida para entornos en los que los proyectos presentan variabilidad significativa y las relaciones entre variables no siguen patrones proporcionales.

La regresión lineal múltiple mostró utilidad en escenarios en los que las relaciones entre variables son más directas, pero su sensibilidad a la dispersión limita su efectividad en conjuntos de datos heterogéneos. Este hallazgo coincide con estudios previos que señalan que la regresión requiere condiciones específicas para ofrecer predicciones precisas. Su inclusión en el análisis permitió confirmar que, aunque sigue siendo un modelo relevante, su aplicación debe considerar la estructura de los datos y la posible presencia de desviaciones.

El comportamiento de las *Support Vector Machines* evidenció la importancia de la normalización y del ajuste de parámetros. El modelo mostró un desempeño intermedio, lo que sugiere que puede ser una alternativa viable cuando se dispone de datos bien escalados y se cuenta con tiempo para realizar ajustes finos. Su sensibilidad a la selección del kernel y a la escala de las variables reafirma la necesidad de un proceso cuidadoso de preparación de los datos.

El método *K-Nearest Neighbors* mostró limitaciones asociadas a la distribución de los datos y a la presencia de valores extremos. Aunque su lógica basada en similitud es intuitiva, su desempeño depende de la estructura del conjunto de datos y de la distancia empleada. Los resultados obtenidos permiten señalar que este modelo puede ser útil en escenarios con agrupaciones naturales, pero requiere condiciones específicas para ofrecer predicciones consistentes.

El estudio también permitió identificar elementos metodológicos que influyen directamente en la calidad de las predicciones. La selección de características se consolidó como un proceso fundamental para mejorar el desempeño de los modelos, especialmente en aquellos que incorporan mecanismos internos para evaluar la importancia de las variables. El tratamiento del desequilibrio resultó determinante para la estabilidad de los resultados, lo que refuerza la

necesidad de aplicar procedimientos de normalización y depuración antes de entrenar los modelos.

Los hallazgos aportan evidencia que puede utilizarse por equipos de desarrollo e investigadores para seleccionar los modelos predictivos más adecuados según las características de sus datos. El estudio contribuye a la línea de investigación sobre la estimación del esfuerzo al ofrecer un análisis comparativo estructurado, sustentado en métricas reconocidas y en procedimientos metodológicos replicables. Además, abre oportunidades para trabajos futuros que incorporen modelos híbridos, redes neuronales o enfoques que integren técnicas de aprendizaje profundo con métodos tradicionales.

La pertinencia del estudio se relaciona con la necesidad de mejorar la precisión en la planificación de proyectos de ingeniería de software, lo que influye directamente en la gestión de recursos, la organización del trabajo y la calidad de los productos entregados. Los resultados obtenidos permiten avanzar hacia prácticas más informadas y la adopción de herramientas analíticas que respondan a la complejidad del desarrollo contemporáneo.

REFERENCIAS

Ahmad, F. B., & Ibrahim, L. M. (2021). *Software Development Effort Estimation Techniques: A Survey*. Journal of Education in Science, Environment and Health, 30(5), 80–92.

Albrecht, A., & Gaffney, J. (2015). *Software function, source lines of code, and effort estimation: A re-evaluation*. IEEE Transactions on Software Engineering, Vol. SE-9, No. 6, pp. 639-648. https://doi.org/10.1109/TSE.1983.235271

Ali, S. S., Ren, J., Zhang, K., Wu, J., & Liu, C. (2023). *Modelo de conjunto heterogéneo para optimizar la precisión de estimación del esfuerzo por software*. IEEE Access, 11, 1–20.

Barbosa dos Santos, R., & Galvão Martins, L. E. (2026). *Técnicas de inteligencia artificial para mejorar estimaciones de costes, esfuerzo y calendario en proyectos de desarrollo de software: Una revisión sistemática de la literatura*. Journal of Software: Evolution and Process. https://doi.org/10.1002/smr.70158

Bou Nassif, A., Ho, D., & Capretz, L. F. (2011). *Modelo de regresión para la estimación del esfuerzo de software basado en el método de puntos de uso*. En Proceedings of the International Conference on Computer and Software Modeling (Vol. 14).

Fernandez-Diego, M., Mendez, E., González-Ladrón-de-Guevara, F., & Abrahão, S. (2020). *Una actualización sobre la estimación del esfuerzo en el desarrollo ágil de software: una revisión sistemática de la literatura*. IEEE Access, 8, 166768–166800. https://doi.org/10.1109/ACCESS.2020.3021664

Jørgensen, M. (2014). *A review of studies on expert estimation of software development effort*. Journal of Systems and Software, 70(1–2), 37–60. https://doi.org/10.1016/S0164-1212(02)00156-5

Kocaguneli, E., & Menzies, T. (2013). *Los modelos de esfuerzo por software deben evaluarse mediante validación leave-one-out*. Journal of Systems and Software, 86(7), 1879–1890. https://doi.org/10.1016/j.jss.2013.02.053

Malik, S., Hamid, M., & Saleem, M. (2026). *A hybrid deep learning model for user story effort estimation*. PLOS One, 21(7), e0353348. https://doi.org/10.1371/journal.pone.0353348

Mittas, N., & Angelis, L. (2013). *Ranking and clustering software cost estimation models through a multiple comparisons algorithm*. IEEE Transactions on Software Engineering, 39(4), 537–551. https://doi.org/10.1109/TSE.2012.45

Rahman, M., Sarwar, H., Kader, M. A., & Gonçalves, T. (2024). *Revisión y análisis empírico de la estimación del esfuerzo de software basada en aprendizaje automático*. IEEE Access, 12, 85661–85680. https://doi.org/10.1109/ACCESS.2024.3404879

Rajput, Y., Razi, M. H., & Sharma, A. K. (2025). *Un análisis comparativo de diferentes técnicas de aprendizaje automático utilizadas en la estimación del esfuerzo por software*. En Proceedings of the 2nd International Conference on Computational Intelligence,

Communication Technology and Networks (CICTN). https://doi.org/10.1109/CICTN64563.2025.10932574

Shepperd, M. J., & MacDonell, S. G. (2012). *Evaluación de sistemas de predicción en la estimación de proyectos de software*. Information and Software Technology, 54(8), 820–827. https://doi.org/10.1016/j.infsof.2011.12.008

Wen, J., Li, S., Lin, Z., & Hu, Y. (2012). *Revisión sistemática de la literatura sobre modelos de estimación del esfuerzo de desarrollo de software basados en aprendizaje automático*. Information and Software Technology, 54(1), 41–59. https://doi.org/10.1016/j.infsof.2011.09.002

Zhang, K., Wang, X., & Ren, J. (2020). *Mejora de la eficiencia de la estimación del tamaño del software basada en puntos de función con un modelo de aprendizaje profundo*. IEEE Access, 9, 107124–107136. https://doi.org/10.1109/ACCESS.2020.2998581