

ESCUELA DE POSGRADO Y ESTUDIOS CONTINUOS

MAESTRÍA EN INGENIERÍA DE SISTEMAS

Modelo de Machine Learning para la precisión de tiempos de desarrollo de software en una empresa consultora en Lima 2023.

Tesis para optar el grado de **MAESTRO** en:

INGENIERÍA DE SISTEMAS CON MENCIÓN EN GERENCIA DE SISTEMAS DE INFORMACIÓN

Autor

Hugo Jurek Camilo Tumba Burgos

Asesor

Doctor Edmundo Rafael Casavilca Maldonado

<https://orcid.org/0000-0001-8625-9811>

Perú

2023

LÍNEA DE INVESTIGACIÓN

Tecnologías emergentes

SUB - LÍNEA DE INVESTIGACIÓN

Data mining. Machine/deep learning. Internet of things (IoT).

Augmented reality (AR). Smart warehousing

JURADO EVALUADOR

Jurado 1	Dr. Alberto Carlos Mendoza De Los Santos	17434055
Presidente	Nombre Y Apellidos	N.º Dni

Jurado 2	Dr. Ronal Santos Paredes Vargas	09565844
	Nombre Y Apellidos	N.º Dni

Jurado 3	Mg. Karina Cardenas Rodriguez	18169440
	Nombre y Apellidos	N.º DNI

Informe similitud



Página 2 of 83 - Descripción general de integridad

Identificador de la entrega trn:oid::1:3062157463




16% Similitud general

El total combinado de todas las coincidencias, incluidas las fuentes superpuestas, para ca...

Filtrado desde el informe

- Bibliografía
- Texto citado
- Texto mencionado
- Coincidencias menores (menos de 8 palabras)

Fuentes principales

- 15%  Fuentes de Internet
- 0%  Publicaciones
- 10%  Trabajos entregados (trabajos del estudiante)

Resumen

Se evidencia el crecimiento de la industria del software por el despliegue de la virtualidad y muchos factores evolutivos de digitalización. En ese sentido, en la presente investigación se planteó como problema principal la estimación de tiempos de desarrollo de software en la fase de planificación, por lo que se tuvo como objetivo el diseño de un modelo de Machine Learning con procesamiento de lenguaje natural para estimación de tiempos de desarrollo de software.

Se propuso del desarrollo de un modelo de machine Learning en la estimación de tipos de evolutivo, siendo éste un factor para la asignación de tiempo; por eso se tiene las dimensiones de identificación de las variables y preparación de datos, diseño del modelo y selección de la técnica y evaluación del modelo.

Finalmente, se implementó el modelo con 64 comentarios observados y estructurados en un Dataset obteniendo un input para la selección de características, entrenamiento del modelo y evaluación de éste con una fiabilidad de 0.68 que es mejor que la estimación tradicional la cual se realiza con expertos en gestión, pero no en un proyecto específico. El modelo de Machine Learning proporciona directamente la experiencia en vivo.

Abstract

Currently, the growth of the software industry is evident due to the deployment of virtuality and many evolutionary factors of digitization. The fundamental problem of this research is the detection of software development times in the planning phase, which is why it is proposed the objective: Develop a Machine Learning model with natural language processing to estimate software development times.

A machine learning model is proposed in the detection of evolutionary types, this being a factor for time allocation, it has the dimensions of identification of the variables and data preparation, model design and selection of the technique and evaluation of the model.

Finally, the model was implemented with 64 comments observed and structured in a Dataset, obtaining input for the selection of characteristics, training of the model and evaluation of it with a reliability of 0.68, which is better than the traditional estimation, which is carried out with experts in management, but not on a specific project. The Machine Learning model directly providing the live experience.

Dedicatoria y Agradecimientos

A mis padres que me inculcaron en cada paso de mi vida los valores necesarios para seguir creciendo profesionalmente, el apoyo académico y la dedicación en lo correcto, encontraron en mí la mejor de las versiones cuando se acababan mis fuerzas y me alentaron y alientan en cada instante de mi vida.

A mis hermanos por el apoyo profesional y académico en cada paso de esta tesis.

A mis profesores y compañeros debido a que sin ellos no tendría la inspiración académica y la amistad que siempre reconforta y fortalece el entorno académico.

Br. Hugo Jurek Camilo Tumba Burgos

Tabla de contenidos

Línea y Sub Línea de Investigación.....	ii
Jurado Evaluador	iii
Informe Similitud	iv
Resumen	v
Abstract.....	vi
Dedicatoria y agradecimiento.....	vii
Tabla de contenidos.....	viii
Índice de tablas y figuras.....	IX

I.	INTRODUCCIÓN.....	1
I.1.	Realidad problemática	1
I.2.	Pregunta de investigación.....	3
I.2.1.	Pregunta general.....	3
I.2.2.	Preguntas específicas.....	3
I.3.	Objetivos de la investigación	3
I.3.1.	Objetivo general.....	3
I.3.2.	Objetivos específicos.....	4
I.4.	Justificación de la investigación	4
I.5.	Alcance de la investigación	5
II.	MARCO TEÓRICO	6
II.1.	Antecedentes	6
II.1.1.	Antecedentes internacionales	6
II.1.2.	Antecedentes nacionales	8
II.2.	Bases teóricas.....	10
II.3.	Marco conceptual (terminología)	20
III.	HIPÓTESIS.....	22
III.1.	Declaración de hipótesis.....	22
III.1.1.	Hipótesis general.....	22
III.1.2.	Hipótesis específicas	22
III.2.	Operacionalización de variables	23
IV.	DESCRIPCIÓN DE MÉTODOS Y ANÁLISIS	25
IV.1.	Tipo de investigación	25
IV.2.	Nivel de investigación	25
IV.3.	Diseño de investigación	25
IV.4.	Método de investigación	26
IV.5.	Población.....	26
IV.6.	Muestra.....	26
IV.7.	Técnicas de recolección de datos	27
IV.7.1.	Técnica	27

IV.7.2. Instrumento	28
V. RESULTADOS	29
V.1. Identificación del tiempo de desarrollo según los puntos de esfuerzo, el tipo de evolutivo, tareas y subtareas del área de desarrollo de software que mejora en la precisión la estimación de tiempos	37
V.2. Desarrollo de un algoritmo de Machine Learning con un árbol de decisión para detección de tiempos del área de desarrollo de software que mejora en la precisión la estimación de tiempos.....	38
V.3. Evaluación de las variables del sistema de estimación de tiempos a tareas y subtareas en el área de desarrollo de software que mejora en la precisión la estimación de tiempos.....	38
VI. DISCUSIÓN, CONCLUSIONES y RECOMENDACIONES	39
VI.1. Discusión	39
VI.2. Conclusiones	41
VI.3. Recomendaciones	42
Lista de referencias	44
Apéndice	47

Índice de tablas

Tabla 1 Matriz de confusión.	17
Tabla 2 Operacionalización de variables	23
Tabla 3 Tipos de evolutivo	32
Tabla 4 Precisión en la estimación de tiempos usando los indicadores de ML.....	37
Tabla 5 Indicadores del diseño del algoritmo.....	38
Tabla 6 Evaluación de las variables.....	39

Índice de figuras

Figura 1 Gráfico de ML Su	11
Figura 2 Vectores visualizados como direcciones y puntos.....	12
Figura 3 Arquitectura de una red neuronal.....	13
Figura 4 Ejemplo de una curva ROC.....	18
Figura 5 Fases genéricas del ciclo de vida de desarrollo	29
Figura 6 Ventana abierta de Anaconda	30
Figura 7 Jupyter Notebook abierto desde el navegador Chrome	30
Figura 8 Importación de librerías.....	31
Figura 9 Extracción de datos estructurados	31
Figura 10 Adaptación de datos CSV a dataframe	32
Figura 11 Código de vectorización de la data	32
Figura 12 Matriz de características.....	33
Figura 13 Muestra de características relevantes.....	33
Figura 14 Muestra de densidad de características	34
Figura 15 Sobre - muestreo	34
Figura 16 Generación de nuevas características	35
Figura 17 Código del entrenamiento del Árbol de decisión	35
Figura 18 Muestra del gráfico del Árbol de decisión	35
Figura 19 Matriz de confusión del Árbol de decisión	36

I. INTRODUCCIÓN

I.1. Realidad problemática

El desarrollo de software, como es normal en una fábrica de un producto y/o servicio, trae consigo problemáticas en su producción que están presentes en las empresas de consultoría digital alrededor del mundo. Dentro de las etapas más importantes del desarrollo se encuentra la planificación, tal como lo manifiesta Intelequia (2020) que dentro de esta fase se realizan tareas clave que pueden llevar al éxito del proyecto y, de la misma manera al fracaso de este. Podemos apreciar de allí que en sí misma la fase de planificación carga la responsabilidad del éxito o el fracaso del proyecto en visión. También indica Intelequia (2020), que resulta con sobrecostos resolver las estimaciones que no se contemplaran de manera oportuna. Se puede decir que la planificación dentro de los proyectos de software se debe mejorar siempre utilizando herramientas oportunamente para evitar sobrecostos en las etapas posteriores y asegurar el éxito del proyecto. Según Cynotec (2022), 1 de cada 4 proyectos de software fracasan debido a una estimación de tiempos defectuosa.

En América latina se rigen los mismos protocolos de desarrollo de software bajo la realidad del entorno y gana mayor demanda a según LinkedIn (2023) Brasil se encuentra en la posición 30 de 60 países más competitivos en términos de tecnología y desarrollo de software. Por su parte Drees y Zhang (2021) indican que el desarrollo de software en Latinoamérica se desenvuelve con criterios de banda ancha local y bajo la mano de obra profesional como recurso del subcontinente teniendo con ello la banda ancha 5G en progreso. Ello coloca al desarrollo de software y sus problemáticas en el contexto del sector latinoamericano, pero con los mismos pilares a nivel mundial, siendo la computadora y el software una industria globalizada, igualmente que la planificación en

el sector digital tiene la misma importancia que en el resto del mundo. Podemos extraer de ZhingFong y Rivas (2023) donde indica que una causa de fracaso en los proyectos de América Latina es el detalle y el nivel al que se llegue, esto se debe a una mala definición del alcance que lleva a una incompleta gestión, estimación de tiempos y creación de tareas. Esto nos da a entender que una parte importante de la estimación de tiempos es el nivel de detalle, el cual depende directamente de la experiencia del estimador y se respalda con los datos que este tenga sobre proyectos ya realizados, lo que se quiere demostrar en esta tesis es que la estimación de una máquina es más exacta que el de una persona, debido a que una persona no puede llegar al alcance experimental de datos en la misma cantidad del tiempo que una máquina, la cual funciona a velocidad luz.

En el Perú, aunque la tecnología está regulada por el decreto supremo N° 118-2018-PCM que es el más puntual sobre regulaciones tecnológicas en el país, también comprende problemas de desarrollo como la planificación y estimación de tiempos; ello otorga una comprensión de lo horizontal que es la problemática de estimación de tiempos en todos los contextos del desarrollo de software. Según LinkedIn (2023), Perú está por detrás de Brasil en el desarrollo de software y por su parte Asenjo et al. (2017) concluye que dentro del proyecto hay modificaciones dentro de las fases que no se gestionan correctamente y esto lleva a una mala estimación del tiempo son dos de las principales causas por las que no se cumple los tiempos en los proyectos. Podemos entender de los autores que en América Latina y el Perú la estimación de tiempos es un tema muy importante para tratar y el experto en estimación tiene una gran cantidad de datos vividos en proyectos anteriores los que ayudan directamente a que se lleve una adecuada gestión de tiempo y creación de tareas. Se pretende resolver la problemática de precisión en la estimación de tiempos de desarrollo en jira y sea esta un paso en la

inteligencia artificial dedicada al desarrollo de software para seleccionar los tiempos en base a los tipos de evolutivo.

I.2. Pregunta de investigación

I.2.1. Pregunta general

¿De qué manera el desarrollo de un modelo de estimación de tiempos con Machine Learning en el área de Desarrollo de Software mejora la precisión en la estimación de tiempos?

I.2.2. Preguntas específicas

Pregunta específica 1: ¿De qué manera identificar el tiempo de desarrollo según los puntos de esfuerzo, el tipo de evolutivo, tareas y subtareas del área de desarrollo de software mejora la precisión en la estimación de tiempos?

Pregunta específica 2: ¿De qué manera el desarrollo de un algoritmo de machine Learning por refuerzo para detección de tiempos del área de desarrollo de software mejora la precisión en la estimación de tiempos?

Pregunta específica 3: ¿De qué manera dimensionar las variables del sistema de estimación de tiempos a 2 tareas y subtareas en el área de desarrollo de software mejora la precisión en la estimación de tiempos?

I.3. Objetivos de la investigación

I.3.1. Objetivo general

Desarrollar un modelo de Machine Learning en el área de Desarrollo de Software que mejora la precisión en la estimación de tiempos.

I.3.2. Objetivos específicos

Objetivo específico 1: Identificar el tiempo de desarrollo según los puntos de esfuerzo, el tipo de evolutivo, tareas y subtareas del área de desarrollo de software que mejora la precisión en la estimación de tiempos.

Objetivo específico 2: Desarrollar un algoritmo de machine Learning con un árbol de decisión para detección de tiempos del área de desarrollo de software que mejora la precisión en la estimación de tiempos.

Objetivo específico 3: Evaluar las variables del sistema de estimación de tiempos a tareas y subtareas en el área de desarrollo de software que mejora la precisión en la estimación de tiempos.

I.4. **Justificación de la investigación**

Justificación aplicativa o práctica: porque el desarrollo de un modelo de estimación de tiempos con Machine Learning en los procesos del Área de desarrollo de software, en principio propone una mejora en la asignación de horas estimadas a los programadores. En segundo lugar, identifica las variables y sus dimensiones que se aplican para el diseño del algoritmo. Por último, propone un análisis del valor agregado en el algoritmo diseñado.

Justificación valorativa: porque el desarrollo de un modelo de estimación de tiempos con Machine Learning en los procesos del Área de desarrollo de software genera una base metodológica para la evolución de la empresa y su crecimiento estratégico. Las organizaciones en todo el mundo están aprovechando el uso de la inteligencia artificial para obtener, de sus históricos y sus datos, información importante con la finalidad de reducir costos y aumentar la calidad y la productividad corporativa con una mejor estimación de tiempos.

Justificación académica: Por su contenido pretende ser una fuente metodológica para realizar el desarrollo de un algoritmo eficiente y eficaz usando la inteligencia artificial con Machine Learning en la estimación de tiempos de desarrollo y también por su contenido estadístico para futuras investigaciones al ser una fuente bibliográfica.

I.5. Alcance de la investigación

Con esta investigación aplicada, el estudio se limita a realizar una serie de tickets de prueba con un data set de acuerdo a la realidad de una empresa consultora en el Perú, año 2023. Este periodo de estudio durará un máximo de 12 meses y terminará cuando se hayan demostrado el objetivo de la tesis. Se realizará el análisis para verificar si percibió un cambio con la estimación de tiempos artificial.

En primer Lugar, realizar la identificación de las variables de predicción de tiempos. En segundo lugar, realizar el análisis del dimensionamiento de las variables usando técnicas de ML. En segundo lugar, se utilizará el muestreo de un porcentaje de la población de datos de estimación de tiempos, los cuales se ingresarán a Machine Learning para calcular el nivel de fiabilidad de acuerdo con los índices internacionales que proporciona la técnica de Machine Learning y comparar la mejora en tiempos de puntos de esfuerzo, tipo de evolutivo, tareas y subtareas tienen una mejora porcentual con respecto a la calculada humana. Asimismo, se medirá la factibilidad de entrenamiento, predicción eficaz, validación digital y desempeño del algoritmo. Por último, se obtiene los resultados del análisis como aporte de investigación.

II. MARCO TEÓRICO

II.1. Antecedentes

II.1.1. Antecedentes internacionales

Geoff Keeling (2022) realizó un estudio sobre el uso de Machine Learning que aporta en la toma de decisiones en el cuidado de la salud cuyo objetivo fue estudiar los algoritmos y su desempeño en predicción de tiempos de atención en pacientes en grupos tradicionalmente desfavorecidos y minorías y verificar si excede su tiempo en los mismos. Fue una investigación cuantitativa descriptiva que usó algoritmos con técnicas estadísticas para predecir y ajustar los tiempos reales y los tiempos estimados. En los resultados los investigadores sustentaron el despliegue permisible de Algoritmos de aprendizaje automático en medicina clínica que se adaptan a minorías y personas con desventajas socioeconómicas subpoblaciones. En conclusión los investigadores resolvieron un algoritmo ML probablemente obstaculice los esfuerzos en la atención a los grupos desfavorecidos y a las minorías, pero es muy efectivo para predicción, lo cual indica que se puede utilizar en áreas de laboratorio o áreas de estudio o gestión.

Jiménez y Naranjo (2020) realizó un estudio sobre la detección de la depresión en línea con NLP y Machine cuyo objetivo fue analizar las características que involucran al autor del texto escrito considerarse como persona deprimida. Fue una investigación cuantitativa que extrajo información de las redes sociales para producir un modelo de Machine Learning. Los resultados fueron exitosos pasando en todos el 80% de fiabilidad pudiendo mejorar la detección y ayuda a personas con síntomas de depresión. En conclusión, el investigador muestra que los resultados mejoran la detección de depresión en todas sus medidas.

Gila Briñas (2021) realizó una tesis para estimar los tiempos de los procesos financieros en el área de planificación con Machine Learning cuyo objetivo fue analizar los procesos financieros y predecir el tiempo en minutos hasta la resolución de una operación para compra de automóviles. Fue una investigación cuantitativa y extrajo información de una entidad financiera para añadir etiquetas y predecir el tiempo del proceso financiero. Los resultados fueron una buena base para plantear un proyecto que se enfoque en la conceptualización de Machine Learning en los procesos financieros teniendo un percentil de 80 en adelante con demoras de entre 20 y 50 minutos. En conclusión, la investigadora con pocos datos fue capaz de predecir la demora de una tarea.

Almache et al. (2021) realizaron un modelo neuronal de estimación para el esfuerzo de Desarrollo en Proyectos de Software (MONEPS) cuyo objetivo fue la estimación en la planificación para el esfuerzo del desarrollo de software validando con el juicio de experto. Fue una investigación cuantitativa donde se midieron las estimaciones de los datos con la predicción una vez ejecutada en costos y en tiempos. Los resultados fueron satisfactorios comparados con el software Cocomo II, simplificando los resultados en costos y operaciones. En conclusión, el modelo de red neuronal aprendió rápido a configurar patrones de comportamiento y a predecir en base a ello y las estimaciones fueron bastante cercanas.

Schulze (2020) realizó un proyecto para la empresa Singular Consulting S.L. sobre el desarrollo de software inteligente para la estimación de tiempos de proyectos, el objetivo de este proyecto fue utilizar las herramientas estadísticas y tecnología de datos para inteligencia de negocios dedicada a los proyectos para calcular el mejor porcentaje de entrenamiento. El diseño de la investigación fue experimental. Los resultados de esta investigación fueron 30% de datos de predicción y 70% de datos de de entrenamiento obteniendo buenos resultados. En conclusión las tecnologías de predicción de datos

como BI y estadística predictiva sí realizan buenas operaciones en conjunto para su estudio.

II.1.2. Antecedentes nacionales

Araujo et al. (2023) realizaron un estudio sobre el uso de algoritmos para la predicción del resultado de los partidos de fútbol utilizando técnicas de la minería de datos, el objetivo de su investigación fue aportar una guía teórica para el uso de Machine Learning. El diseño del estudio fue experimental. Este estudio mostró que las técnicas Machine Learning más usadas a nivel global son el aprendizaje supervisado y el aprendizaje no. En conclusión, el estudio propuso un modelo innovador y rápido para la predicción del resultado de los partidos de fútbol, apoyado con Data Mining centrado en Machine Learning.

Sanchez y Cruz (2023) realizaron una simulación de la estimación de tiempos variables de acarreo y movimiento de tierras utilizando el software TALPAC, el objetivo fue optimizar la gestión del tiempo con tecnologías de software. el diseño es de estudio experimental. Los resultados fueron óptimos teniendo una reducción en la estimación de tiempos. En conclusión, se pudo optimizar la gestión de tiempos con el software TALPAC.

Villegas (2021) realizó un estudio sobre el uso de un algoritmo de Machine Learning para detectar sitios web phishing, el objetivo de esta investigación comprobar que los algoritmos de Machine Learning mejoran el rendimiento de detección de sitios phishing. Fue una investigación cuantitativa. Los resultados mostraron que los algoritmos de Machine Learning representó una mejora de 1.55% que llegó a 95.35%, lo cual representa un mejor rendimiento frente a 93.80% de investigaciones anteriores. En

conclusión, el estudio se propone un modelo novedoso y eficiente para predecir los sitios web phishing con Machine Learning.

Davila y Davila (2021) desarrollaron una aplicación para el análisis del mercado, basado en el procesamiento del lenguaje natural sobre la plataforma de red social Twitter y usaron técnicas de Machine Learning para predecir si el lanzamiento de un nuevo producto será exitoso. El objetivo era comprobar el uso de herramientas de IA como Machine Learning para predecir el éxito de lanzamiento de nuevos productos. Fue una investigación cuantitativa. Los resultados mostraron que el camino a la predicción está guiado y es muy posible mejorar la detección de éxito. En conclusión, se pudo mejorar operativamente la detección automática del éxito de nuevos lanzamientos, pero la eficacia está en vías de mejorar.

Chang (2023) desarrolló una tesis de grado donde utiliza las técnicas de machine learning para estimar el dinero en adquisiciones públicas dentro del territorio peruano. El objetivo fue realizar un modelo de machine Learning para estimar los costos de las adquisiciones en entidades públicas de manera eficiente. Fue una investigación cuantitativa. Los resultados tras las pruebas de desempeño mostraron que es mejor utilizar las técnicas de regresión lineal múltiple para estimación de costos porque mejora la eficacia.

II.2. Bases teóricas

Machine Learning

Los algoritmos de aprendizaje automático son una rama de la inteligencia artificial que está extendida en diferentes campos para resolver un problema práctico y se pueden entender mejor recurriendo a la teoría de ella. Por ejemplo, Burkov (2019) indica que Machine Learning se puede expresar como un algoritmo que busca una respuesta en base a la recopilación de diferentes datos agrupados y estructurados que sirven para desarrollar un modelo y encontrar respuestas con ejemplos reales que se integrarán al mismo algoritmo para que este aprenda de ello y obtenga más experiencia. Se puede entender que el algoritmo guarda datos pasados los cuales se entrenan para generar un modelo y en puesta a producción integra también datos nuevos que alimentarán la experiencia de este y exportará un resultado puntual y semejante al real que mejorará para proporcionar resultados más fiables.

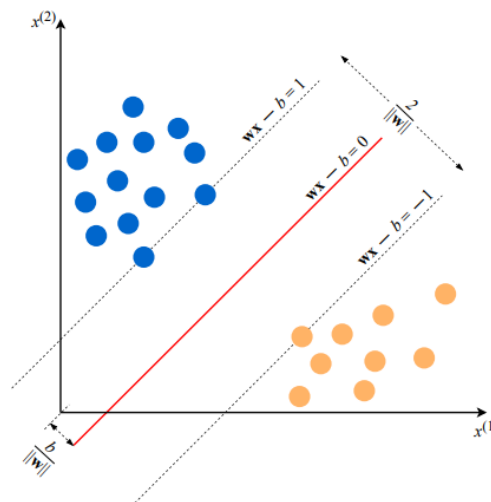
Para entender mejor cómo se comporta debemos dimensionar las 4 técnicas utilizadas, Machine Learning supervisado, Machine Learning sin supervisar, Aprendizaje semi supervisado, Machine Learning por refuerzo, cada una de ellas con diferencias en el desarrollo del modelo que se usan en diferentes ámbitos y dependerán del criterio del experto humano para elegir la más fiable dependiendo de la cantidad de datos que se tenga, los recursos humanos y el contexto demográfico, social y problemática a la que se enfrente. se describe las técnicas utilizadas:

Machine Learning supervisado

Son los algoritmos que comprenden data verificada y estructurada. Por ejemplo, Burkov (2019) indica que el objetivo de un algoritmo Machine Learning bajo supervisión es utilizar una colección de datos estructurados bajo x , y , etc, tomar ese vector de

características como entrada y proporcionar como resultado una deducción del siguiente dato o datos. Podemos entender de esto que los datos se agrupan para solucionar un problema y pueden ser ingresados a la máquina generando secuencias de comportamiento que facilitarán el resultado. Las pruebas y verificación son muy accesibles para el desarrollador, entregándole mejor control sobre los cambios en la data y los resultados.

Figura 1
Gráfico de ML Su



Nota. Tomado de *The Hundred-Page Machine Learning Book* (p.7), por A. Burkov, 2019, Draft.

Un ejemplo de ello se puede ver en la data que se ingresa para calcular el peso de una fruta, si se puede estructurar el estado del color y los calibres en un vector, el algoritmo puede encontrar el comportamiento y mostrarlo en un gráfico lineal, el cual correlacionará eficientemente los calibres y su estado del color con el peso de vegetal y mostrar un resultado semejante al real.

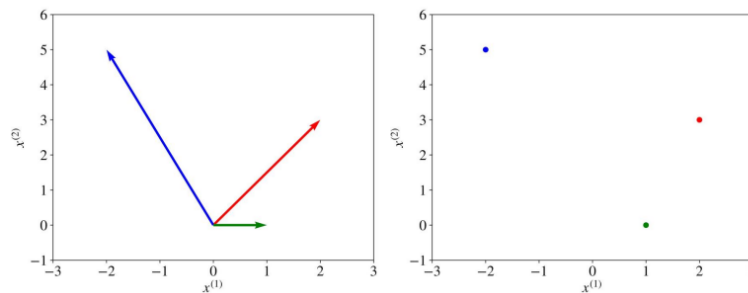
Machine Learning sin supervisión

Son los algoritmos que comprenden data no verificada y se agrupan para solucionar un problema. Burkov (2019) indica que se espera que el uso de mucha data no estructurada

pueda ayudar al algoritmo a producir un mejor modelo. Lo que nos quiere decir Burkov (2019) es que este algoritmo al contener data no supervisada trae una mayor longitud de vector y es la cantidad de datos que se ingresan debido a que no se requiere además del preprocesamiento humano supervisado y no limita la cantidad de data, pero no mejora las pruebas siendo estas visibles de manera genérica para el desarrollador. Estos patrones deben ser analizados por la máquina sin tener la deficiencia de la cantidad de datos. No se tendrá los resultados finales en sí mismos, pero la máquina entenderá los patrones de comportamiento que existen y propondrá un resultado óptimo.

Figura 2

Vectores visualizados como direcciones y puntos.



Nota. Tomado de *The Hundred-Page Machine Learning Book* (p.12), por A. Burkov, 2019, Draft.

Un ejemplo de ello puede ser el de llamadas de spam, se puede caracterizar el comportamiento de un individuo por su tono, risas, si contiene palabras explícitas o el tiempo de llamadas, que un humano puede captar rápidamente y enseñar a la máquina.

Aprendizaje semi supervisado

Esta técnica se compone por dos tipos de datos, supervisados y sin supervisar cuyo objetivo es dar respuestas viables cuando hay grandes cantidades de datos no procesados ni estructurados. Hurwitz y Kirsch (2018) indican que, al añadir data no estructurada, parece que se agrega mayor incertidumbre, sin embargo, cuando se agrega más ejemplos al vector en una muestra más grande dibuja mejor el gráfico de

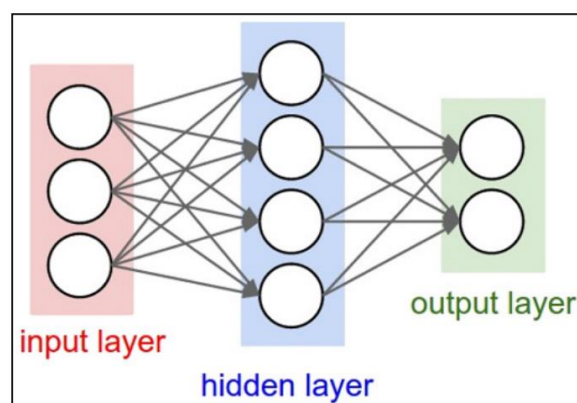
distribución resultante. Ellos indican que el algoritmo consiste en ingresar cantidades de datos etiquetados para aumentar los grupos de datos sin etiquetar. Los datos etiquetados influyen en dar más experiencia a la máquina y funcionamiento al sistema y mejorar considerablemente el tiempo de respuesta y fiabilidad del aprendizaje.

Machine Learning por refuerzo

En contexto en el aprendizaje supervisado la máquina recibe la respuesta real y aprende buscando correlaciones entre los resultados en cambio el modelo de aprendizaje por refuerzo no incluye una respuesta de real, pero ingresa un conjunto de patrones potenciales. Hurwitz y Kirsch (2018) indican sobre este que es un modelo conductual y obtiene aprendizaje del análisis de los datos y es guiado a un mejor resultado y en este caso la máquina aprende por prueba y error. Podemos apreciar de los autores que en este caso no aprenderán de los datos, más bien usarán esto con los comportamientos de prueba y error y reforzarán los ejemplos atinados, de esta manera se entrena a la máquina sin esperar que con el conjunto de datos exporten resultados.

Figura 3

Arquitectura de una red neuronal



Nota. Adaptado de *Machine Learning for dummies* (p.31), por J. Hurwitz y D. Kirsch, 2019, IBL Limited Edition.

Un ejemplo de esto puede ser enseñarle a la máquina a jugar ludo que no se podrá enseñar todas las maneras en las que se puede mover la ficha y los dados, pero sí puede

obtener recompensa al reiniciar a los oponentes o llegar más veces al final del camino en el tablero.

Procesamiento de lenguaje natural

La comunicación entre seres humanos en un proceso muy simple para las personas y a la vez muy complejo en su análisis tanto es así que por la experiencia humana se puede obtener información desde el habla oral y escrita hasta los gestos e íconos que se utilizan. De esta manera el ser humano enfocado en su estructura ha creado y desarrollado instituciones dedicadas exclusivamente al estudio de la gramática en todas partes del mundo, por poner unos ejemplos, la Real Academia Española, Instituto Digital César Chávez para el español en Norteamérica, Centro de Estudios Lingüísticos y Literarios de México, la Academia peruana de la lengua en Lima.

Por su parte las ciencias de la computación incurrieron en la necesidad de crear algoritmos que puedan analizar la creatividad humana en el lenguaje y poder extraer sentimientos, decisiones, ideas, acciones, de esta manera se tiene a los algoritmos de procesamiento del lenguaje natural, tal como lo indica Villegas (2021) la máquina debe contar con la capacidad de comprensión de jergas, errores, hábitos propios del ser humano que no se rigen a un sí o un no sino más bien al mundo de las semejanzas y probabilidades, de esta manera, tenemos las siguientes técnicas para analizar textos:

Basado en la frecuencia de aparición. Tal como lo indican Jiménez y Naranjo (2020) indicando que este algoritmo se enfoca en el número de apariciones de palabras dentro de un texto ingresado. De esta manera extraer la probabilidad de aparición y predecir un resultado.

Bolsa de palabras (BOW). Sobre este algoritmo Jiménez y Naranjo (2020) indican que cuenta las palabras diferentes de un texto y analiza la frecuencia de las palabras

enumeradas del vocabulario en una lista vectorizada. Existe un inconveniente con este algoritmo y es que al aparecer nuevas palabras que no están en el vocabulario inicial estas no tendrán una representación en la lista codificada.

Término Frecuencia-Frecuencia Inversa de Documento (TF-IDF). Este algoritmo representa estadísticamente la importancia de una palabra con respecto a un documento en un grupo de documentos. Jiménez y Naranjo (2020) añaden que la puntuación TF-IDF de una palabra en un documento dentro de un conjunto de documentos, con tamaño N es calculado por la siguiente ecuación:

$$tf\ idf(w, d, D) = tf(w, d) * idf(w, D) \quad (1)$$

Ponderando la frecuencia de una palabra dentro de un documento (tf)

$$tf(w, d) = \log(1 + freq(w, d)) \quad (2)$$

La frecuencia de una palabra dentro de todos los documentos (idf):

$$idf(w, D) = \log\left(\frac{N}{count(d \in D: w \in d)}\right) \quad (3)$$

N-Grams. Esta técnica a diferencia de la cuenta de palabras utiliza la frecuencia de grupos de palabras que aparecen en una secuencia. La técnica BOW también podría denominarse 1 gramo o un-gramo. Jiménez y Naranjo (2020) indican que este algoritmo se puede usar para encontrar expresiones más comunes en un texto o grupo de textos.

Algoritmos basados en el contexto. Una forma más natural de abordar la comprensión del lenguaje es analizar el contexto de la palabra para ver sus relaciones con ella. Se muestra los siguientes, incrustación de palabras e incrustación de documentos:

Incrustación de palabras. Este algoritmo permite comprender palabras con significado similar. Jiménez y Naranjo (2020) indican que esta técnica abstrae las palabras en

vectores similares a otros utilizados en el mismo contexto. Un ejemplo dinámico es el de la palabra perro y gato que son similares al ser ambos significados de animal. Otra manera de ejemplificarlo es con el vector de la palabra verde el cual estará más cerca del vector de la palabra pepino que del vector de la palabra tomate, ya que refiere a una característica principal de él. Algunas herramientas de incrustación de palabras más utilizadas son Word2Vec, un algoritmo de red neuronal superficial para aprender incrustaciones de palabras, fastText y GloVe.

Incrustación de documentos. Se puede aplicar un enfoque similar al de la incrustación de palabras seguido para codificar textos completos. Es un enfoque más complejo que palabras o frases así lo indica Jiménez y Naranjo (2020) en su estudio mencionan que mientras Word2Vec aprende a proyectar palabras en un espacio de salida de n dimensiones, Paragraph Vector (o Doc2Vec) hace lo mismo con párrafos o textos completos.

Técnicas de vectorización de NLP

Vectorización es la denominación que se le asigna a la abstracción que refiere estructurar el lenguaje natural para convertirlo en números reales, que es el formato que admiten los modelos ML. Con esto podemos resumir en un vector lista, las palabras más mencionadas en un mismo objetivo, con ello podemos asignar una probabilidad de aparecer y analizar si está inclinado por un resultado u otro.

Medición de desempeño del algoritmo de Machine Learning

Se cuenta con una significativa cantidad de métricas para medir el desempeño por lo tanto se tiene que seleccionar cuáles son las métricas más adecuadas para el algoritmo utilizado.

Matriz de confusión. Esta tabla recoge las frecuencias de valores positivos y negativos y es una de las métricas más intuitivas y descriptivas que se utilizan para encontrar la precisión y corrección de un algoritmo de aprendizaje automático. Su uso principal es en problemas de clasificación.

Dónde:

Precisión en la estimación de tiempos usando diferentes algoritmos

Tabla 1
Matriz de confusión.

Clases	Predicción de positivos	Predicción de negativos
Positivos reales	Verdaderos Positivos (VP)	Falsos Negativos (FN)
Negativos reales	Falsos Positivos (FP)	Verdaderos Negativos (VN)

Nota. Adaptado de *Matriz de confusión* (p.30), por J. Villegas, 2021, Universidad señor de Sipán.

VP: Verdaderos positivos, es el número correcto de predicciones positivas.

FP: Falsos Positivos, es el número incorrecto de predicciones positivas.

FN: Falsos Negativos, es el número incorrecto de predicciones negativas.

VN: Verdaderos Negativos, es el número correcto de predicciones negativas.

Con los datos de la matriz de confusión se puede extraer los siguientes indicadores:

Accuracy – exactitud: Es comúnmente empleado y posiblemente la principal elección para evaluar el rendimiento de un algoritmo en situaciones de clasificación. Se define como la proporción entre la cantidad de elementos clasificados correctamente y el total de observaciones.

$$Accuracy = \frac{VP+VN}{VP+FN+FP+FN} \quad (4)$$

Precisión: Muestra "cuántos elementos de datos seleccionados son pertinentes". En otras palabras, indica de las predicciones positivas realizadas por un algoritmo, cuántas son efectivamente positivas. La precisión se obtiene al dividir el número de verdaderos positivos entre la suma de verdaderos positivos y falsos positivos.

$$\text{Precisión} = \frac{VP}{VP+FP} \quad (5)$$

Recall: Corresponde a la tasa de Verdaderos Positivos y establece la sensibilidad como la proporción de verdaderos positivos, calculada al dividir el número de verdaderos positivos entre la suma de verdaderos positivos y falsos negativos.

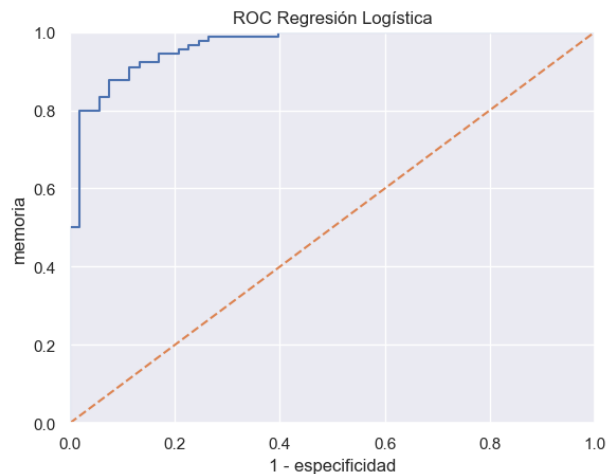
$$\text{Recall} = \frac{VP}{VP+FN} \quad (6)$$

F1-Score. Esta fórmula evalúa el desempeño de un algoritmo considerando tanto la precisión como la sensibilidad; incorpora en su expresión tanto la exactitud como la recall.

$$F1 = 2 \cdot \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}} \quad (7)$$

Curva ROC y AUC. Ilustra el rendimiento del receptor y el área bajo la curva. En la curva, se visualiza la Tasa de Falsos Positivos (TFP) en comparación con la Tasa de Verdaderos Positivos (TVP). La curva ideal se encuentra en la parte superior izquierda, indicando un alto recall y al mismo tiempo, una baja Tasa de Falsos Positivos. Se puede apreciar un ejemplo en la figura 4 extraída directamente del software de Machine Learning en el lenguaje de programación Python.

Figura 4
Ejemplo de una curva ROC



Calidad del Software según el estándar internacional ISO 9126 Según Verity (2022)

Confiabilidad: Evalúa la capacidad y facilidad de recuperación, la capacidad para mitigar fallos, la duración durante la cual el software permanece disponible para su utilización y su nivel de tolerancia. En resumen, cuantifica los posibles fallos que el producto de software podría experimentar.

Usabilidad: Evalúa la facilidad de uso, la interacción del usuario con el sistema, entre otros aspectos. Puede manifestarse como la cantidad de pasos requeridos para llevar a cabo una función.

Eficiencia: Evalúa la eficiencia con la que el software utiliza los recursos del sistema, considerando el tiempo de uso y los recursos disponibles. Puede expresarse como el tiempo necesario en segundos para que el software responda a su funcionalidad central.

Facilidad de mantenimiento: Es la facilidad de modificación de la herramienta, la facilidad para hacerle pruebas de rendimiento y toda la ingeniería de requerimientos. Se puede definir en términos de escalabilidad y mantenimiento que los desarrolladores pueden entender. Se puede decir que un código es entendible cuando está bien documentado, sus porciones de código están bien comentadas.

Predicción en la estimación de tiempos: Se entiende como predicción en la estimación de tiempos a los datos relacionados con tareas que no se han realizado y se requiere una estimación prematura para el análisis de costos. Las empresas resuelven esta predicción con el análisis de líder experto. Según Schulze (2020), este análisis de la predicción de tiempos se realiza por el experto teniendo en consideración el error en base a la cantidad de proyectos que ha estimado. Con esto podemos entender que la predecir los tiempos

nos lleva a variables de estratificación en jira como lo son tipo de evolutivo para estimar un costo de ello.

II.3. Marco conceptual (terminología)

Tiempo de desarrollo de software: Según Verity (2012) es el tiempo que se toma o tomará para realizar una tarea, esta puede ser estimada o ejecutada la cual difiere de si ya se realizó o es una proyección.

Algoritmo de Machine Learning: Según Burkov (2019) son aquellos algoritmos que engloban el Machine Learning.

Biblioteca de clases en Python: Destacando lo anterior sobre Algoritmo de machine Learning de Burkov (2019), son los algoritmos que ya están hechos en el lenguaje de programación Python y se reutilizan para la investigación.

Incidencia: Según Atlassian (2023) es el tipo de tarea que se realizará como defecto, desarrollo nuevo, evolución.

Tarea de desarrollo de software: Según Atlassian (2023) es la parte granular de un proyecto de software que define quién lo desarrollará y en qué tiempo.

Subtarea de desarrollo de software: Según Atlassian (2023) es la parte granular de un proyecto de software que define qué se realizará dentro de una tarea. Es un nivel más específico con respecto a la tarea.

Punto de función de esfuerzo: En el sitio web PMO Informática (2015) define que se estima al inicio del proyecto de software, con ello se indica al cliente el costo y se define la cantidad de tareas, recursos y tiempo que se tomará ese punto en específico en base a las horas por esfuerzo de las tareas.

Árbol de decisión: Según Burkov (2019) es una técnica basada en aprendizaje supervisado el cuál potencia el análisis predictivo con alta precisión, estructurando un conjunto de preguntas graficado en forma de árbol dirigiéndose a un lado o al otro hasta predecir lo que se requiere.

III. HIPÓTESIS

III.1. Declaración de hipótesis

III.1.1. Hipótesis general

El desarrollo de un modelo de estimación de tiempos con Machine Learning en el área de Desarrollo de Software tiene una mejor precisión en la estimación de tiempos.

III.1.2. Hipótesis específicas

Hipótesis específica 1: El tiempo de desarrollo según los puntos de esfuerzo, el tipo de evolutivo, tareas y subtareas del área de desarrollo de software tiene una mejor precisión en la estimación de tiempos.

Hipótesis específica 2: El algoritmo de machine Learning para estimación de tiempos del área de desarrollo de software tiene una mejor precisión en la estimación de tiempos.

Hipótesis específica 3: Las variables del modelo de estimación de tiempos a tareas y subtareas en el área de desarrollo de software tiene una mejor precisión en la estimación de tiempos.

III.2. Operacionalización de variables

Tabla 2
Operacionalización de variables

Variable 1	Tipo de variable según su naturaleza	Definición Conceptual	Definición Operacional	Dimensiones	Indicadores	Nivel De Medición
Modelo de estimación de tiempos con Machine Learning en el área de Desarrollo de Software	Cuantitativa	Según Wan Jiang et al. (2015) Un modelo de estimación de tiempos con Machine Learning son aquellos algoritmos que requiere una función de predicción de alta precisión. Por lo tanto, los enfoques de predicción deben poder predecir el esfuerzo y duración, con alta precisión. Los datos del proyecto predecirán el esfuerzo y la duración futuros del proyecto. Sin embargo, debido a la Aunque existe una gran cantidad de algoritmos de aprendizaje automático	Se entiende por Modelo de estimación de tiempos con Machine Learning como una técnica que utiliza algoritmos de inteligencia artificial para realizar la predicción y estimación de un resultado en base a experiencia previamente ingresada.	Diseño del algoritmo	Confiabilidad Eficiencia Mantenibilidad Usabilidad	Razón

Precisión en la estimación de tiempos.	Cuantitativa	Según Salazar-B (2009) precisión en la estimación del tiempo de desarrollo de software implica la aproximación de un periodo calculado para la finalización de todas las tareas. Estas tareas abarcan diversos conjuntos de actividades necesarias para llevar a cabo el desarrollo de un producto. La estimación global de la tarea varía en función de diversos factores, como la complejidad del proyecto, la cantidad de mano de obra necesaria, la experiencia general del equipo y la disponibilidad de recursos.	Precisión	
		Precisión en la estimación de tiempos	Accuracy	Razón
		Precisión en la estimación de tiempos de desarrollo de software detalla el tiempo que se predice se culminará una tarea o un grupo de tareas.	Recall	
		Evaluación de las variables	F1-Score	

IV. DESCRIPCIÓN DE MÉTODOS Y ANÁLISIS

IV.1. Tipo de investigación

La investigación por su naturaleza es aplicada con pruebas de laboratorio y los datos se medirán cuantitativamente y se validarán de la misma manera. Hernández y Mendoza (2018) nos indica que en el caso del tipo de investigación cuantitativa se desarrolla un diseño experimental para validar las hipótesis formuladas. Se cuenta con datos ingresados a un modelo de Machine Learning diseñado durante la investigación y con experimentación se validará las hipótesis, asimismo de enfoque cuantitativo.

IV.2. Nivel de investigación

La investigación es de nivel aplicativo. Podemos analizar de Vara (2012) que la investigación es práctica y se basa en resultados como solución de un problema es por naturaleza aplicada. Esta investigación tiene como aporte la aplicación los algoritmos computacionales de Machine Learning con los cuales se evaluará la solución al problema de estimación de tiempos en las empresas consultoras de software.

IV.3. Diseño de investigación

La investigación es experimental porque se utiliza una muestra de la población y esta no se altera, solo reflejada en el tiempo y depende del algoritmo propuesto. Se usa el Machine Learning para resolver el problema de Estimación de tiempos, así mismo se obtendrán los datos y se ajustarán a lo que requiere la investigación, probar que Machine Learning mejora la estimación de tiempos. Hernández y Mendoza (2018) nos indica que las investigaciones que utilizan un diseño experimental recolectan los datos y validan los mismos con la experimentación.

IV.4. **Método de investigación**

Se usará el método hipotético deductivo, debido a que se obtiene una hipótesis y pregunta de investigación con conclusiones en base a las premisas analizadas de los datos empíricos, en este caso las variables de estimación, los cuales se ejecutarán como una síntesis en el sistema de Machine Learning.

Podemos analizar que Rodríguez y Pérez (2017) nos indica que el método hipotético deductivo nos ayuda a responder claramente la pregunta de investigación con los resultados obtenidos los cuales servirán para aportar conocimiento. Machine Learning busca descomponer y analizar la información en sus partes como variables, las cuales servirán como fuente de data para el modelo que se aportará a las predicciones.

IV.5. **Población**

La población es el conjunto de datos de tareas estimadas y ejecutadas de una consultora de software, validadas por un experto en Machine Learning (Anexo VIII), de un Dataset extraída de una empresa del rubro.

Cada sprint quincenal contiene las tareas nuevas y el rango de investigación es de las tareas realizadas al 2023. En todo el 2023 en el Dataset validado tiene hasta agosto 16 Sprint, con una población de 64 tareas. El instrumento y la población validada por el juicio de experto. Se puede revisar el anexo VI, VII y VIII.

IV.6. **Muestra**

Conjunto unitario estimado de datos de tareas consideradas y ejecutadas de una consultora de software. Según Vara (2012) el tamaño de la muestra va a depender directamente de la precisión requerida.

Aplicando criterios de exclusión con respecto a tareas adaptables al modelo que contengan el tiempo definido inicialmente antes del desarrollo de la tarea, ya que se define durante o después del desarrollo la precisión humana aumenta debido al conocimiento previsto. Usaremos una muestra de 34 tareas a evaluar de un total de 64 de un data set que cumplen con el criterio de exclusión.

IV.7. **Técnicas de recolección de datos**

IV.7.1. **Técnica**

Se utilizará dos técnicas para recolectar datos y validarlos, documentación y observación. Podemos recoger de Hernández y Mendoza (2018) que los tipos de investigación cualitativa utilizan técnicas para recolectar datos y explica que la documentación y observación son muy utilizadas en las investigaciones de experimentación. Esta investigación es de tipo cuantitativa como medición del modelo de Machine Learning que recolectará los datos con la documentación digitalizada y observará los experimentos de laboratorio.

A continuación, se detalla cada técnica.

Documentación: Se recolecta información de una base de datos de desarrollo de software donde se visualiza las distintas variables como lo son Puntos de esfuerzo, Tipo de evolutivo, tareas y subtareas ejecutadas y estimadas, las cuales se pondrán a prueba con las reales para verificar la fiabilidad del algoritmo de machine Learning.

Observación: Se recogerán los datos de la experimentación y se visualizará cuánto tiempo toma el sistema con el algoritmo de Machine Learning con su contraparte humana para verificar la eficiencia de tiempo, costo y eficacia en calidad de predicción esto se basa en que a menos desviación del resultado mejor calidad de predicción.

IV.7.2. Instrumento

Información de tiempos de desarrollo en jira: Aquí se encuentra los datos obtenidos de una consultora de software de Lima donde se detalla los campos: Tipo de evolutivo, tareas y subtareas ejecutadas y estimadas. (Anexo IV)

Como instrumento utilizaremos la guía de observación para realizar una observación estructurada de los experimentos que se realizarán. Según Rodríguez y Pérez (2017) la observación estructurada es una técnica cuantitativa y es utilizada para registrar el fenómeno estudiado de forma sistemática y directa. En esta investigación se observará los experimentos y se estudiarán los resultados.

A continuación, se detalla el instrumento.

Guía de observación: Se realizará pruebas al algoritmo de Machine Learning y se detallará en una guía de observación a cada análisis para medir su probabilidad y nivel de mejora de estimación de tiempo. Se puede ver en el Anexo II la guía de observación.

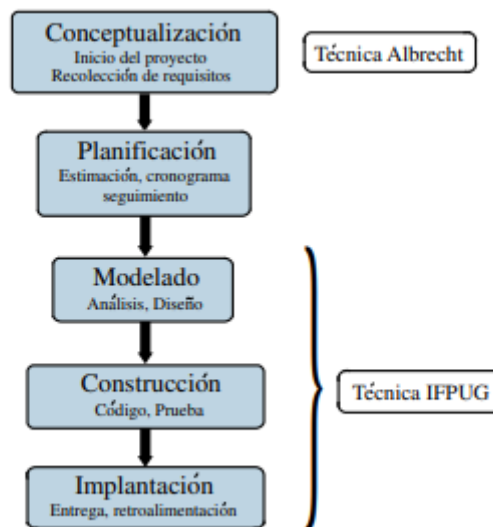
Guía de observación: Se realizará pruebas al algoritmo de Machine Learning y se detallará en una guía de observación a cada análisis para medir su probabilidad y nivel de mejora de estimación de tiempo. Se puede ver en el Anexo III la guía de observación.

V. RESULTADOS

La presente investigación como objetivo general planteó desarrollar un modelo de Machine Learning en el área de Desarrollo de Software que mejora la precisión en la estimación de tiempos.

El modelo realizado está enfocado en la fase de planificación dentro de la actividad de estimación, cronograma y seguimiento, donde se estima los tiempos del desarrollo tal como se puede visualizar en la Figura 5. Este modelo de predicción de tiempos sigue la metodología descrita anteriormente. Se codifica en el lenguaje de programación Python, en un cuaderno de Jupyter Notebook en Anaconda. Se hace uso de las librerías open source que se pueden descargar desde la página oficial <https://anaconda.org/>.

Figura 5
Fases genéricas del ciclo de vida de desarrollo



Nota. Tomado de *Estimación de proyectos de software* (p.128), por G. Salazar, 2009, Universidad EAFIT.

En la Figura 6 podemos apreciar a la herramienta de libre llamada Anaconda tal como se puede apreciar en la Figura 6, desde donde abriremos Jupyter Notebook en la Figura 7 y desarrollaremos el algoritmo en el lenguaje de programación Python.

Figura 6
Ventana abierta de Anaconda

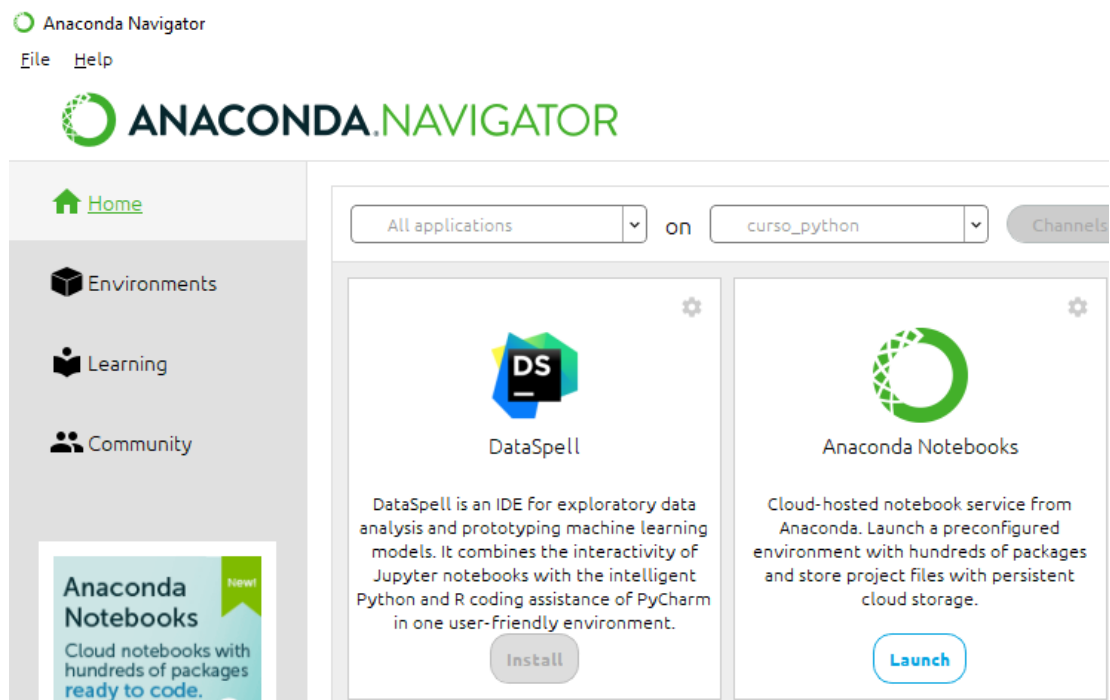
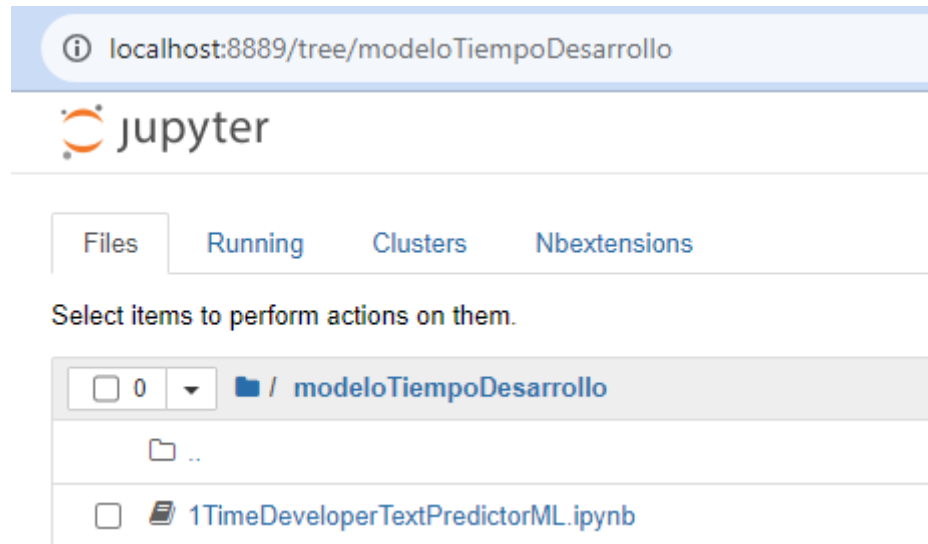


Figura 7
Jupyter Notebook abierto desde el navegador Chrome



Se usa las librerías SKLEARN y PANDAS las cuales ya tienen el aporte de la optimización de código en Python y abrevia mucho tiempo de desarrollo estas colocan código implícito que engloba muchas técnicas de Machine Learning podemos visualizar en la Figura 8.

Figura 8

Importación de librerías

```
#Paquetes para Lectura y recuento
from sklearn.datasets import fetch_20newsgroups
import pandas as pd
#Si lo que queremos es Leer nuestros CSV...
import pandas as pd
```

El primer paso en el código es extraer la data desde una BD donde se encuentra los 32 ítems señalados en la muestra, para razones prácticas del estudio se extrae desde la herramienta CSV donde se encuentra el Dataset con los comentarios y su respectivo resultado, el cuál es el Tipo de Evolutivo véase la Figura 9.

Figura 9

Extracción de datos estructurados

```
dataJalada = pd.read_csv('dataML.csv'
                        , sep = ';', encoding='latin-1'
                        , header=None, names=['data','tipo'])
```

El objetivo del estudio es introducir un comentario cualquiera de alguna solicitud de un cliente y el modelo de Machine Learning pueda predecir qué tipo de evolutivo es, siguiendo un patrón de características de texto utilizando NLP de ML. Podemos apreciar en la figura 10 la [data], que vendría a contener los comentarios de evolutivos que ya tienen un [tipo], este tipo tiene 3 posibles evolutivos, evolutivo tipo incidente, nuevo desarrollo o modificación del desarrollo. Para que se pueda extraer características de los comentarios, es necesario vectorizar las palabras realizando una cuenta de las veces en las que aparece como en la Figura 11, para tener valores objetivos con los cuales se puede discernir entre los tipos de evolutivo.

Figura 10

Adaptación de datos CSV a dataframe

```
#Pasamos Los datos a un DataFrame (datos en pandas)
data = pd.DataFrame(dataJalada, columns = ['data'])
target = pd.DataFrame(dataJalada, columns = ['tipo'])
```

Figura 11

Código de vectorización de la data

```
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.feature_extraction.text import TfidfVectorizer

#Inicialización de los vectores de extracción de características
#CountVectorizer
vectCV = CountVectorizer(max_features = 400
                        ,stop_words=['0000000000003906', '001', '002', '03492', '04', '05', '06', '062', '07', '08',
                                     '09', '10', '11', '12', '13', '14', '15', '16', '17', '18', '19', '20', '21', '22', '23', '24', '25', '26', '27', '28', '29', '30', '31', '32', '33', '34', '35', '36', '37', '38', '39', '40', '41', '42', '43', '44', '45', '46', '47', '48', '49', '50', '51', '52', '53', '54', '55', '56', '57', '58', '59', '60', '61', '62', '63', '64', '65', '66', '67', '68', '69', '70', '71', '72', '73', '74', '75', '76', '77', '78', '79', '80', '81', '82', '83', '84', '85', '86', '87', '88', '89', '90', '91', '92', '93', '94', '95', '96', '97', '98', '99'])
```

Podemos apreciar los tipos de resultado que va a predecir el modelo en la Tabla 3, estos son elaborados por el proyecto de software al que estén enfocados y configurados en el modelo. A manera de laboratorio se analizan estos 3 tipos que son genéricos para todos los proyectos de software.

Tabla 3

Tipos de evolutivo

Tipos de evolutivo	Descripción
--------------------	-------------

Incidente de defectos	Se categoriza con el número 5 en el sistema y es reportado por el cliente o de manera externa o de manera interna por los desarrolladores en pruebas cruzadas, se estima por defecto en resolución de incidentes de una hora con solo una dimensión.
Nuevo desarrollo	Se categoriza con el número 1 y es solicitado por el cliente para desarrollar e incrementar la interfaz con nuevas pantallas y funcionalidades, se estima tiempos de 4 horas por dimensión con 3 dimensiones como mínimo correspondientes a desarrollo Back-End, Front-End y Pruebas.
Modificación de lo desarrollado	Se categoriza con el número 4 y es solicitado por el cliente para modificar la interfaz agrega o quitar funcionalidades, se estima tiempos de 4 horas por dimensión con 1 dimensión como mínimo, dependiendo de la modificación, esta puede estar localizada en desarrollo Back-End, Front-End.

La vectorización de la variable cuenta la cantidad de veces que aparece una palabra dentro de un comentario y revisa de qué manera se repite en los demás comentarios y de esta manera abstraer un texto, tal como podemos ver en la Figura 12. De la misma manera se puede mostrar las características relevantes como en la Figura 13.

Figura 12

Matriz de características

```
#Obtención de Las matrices de características
X_vectorizedCV = vectCV.transform(data['data'])
```

Figura 13

Muestra de características relevantes

```
print(vectCV.get_feature_names_out())
['actualización' 'actualizar' 'adjunta' 'adjuntamos' 'adjuntan' 'adjunto'
'adjuntos' 'detección' 'detectado' 'documentación' 'documento'
'documentos' 'error' 'errores' 'falla' 'fallo' 'falta' 'faltan'
'incidencia' 'normalizadas' 'normalizado' 'normalizados' 'pdf' 'probar'
'problema' 'reabrirse' 'reapertura' 'renormalizar' 'subsanan']
```

Podemos revisar la cantidad de veces que se repiten las palabras entre los diferentes textos teniendo como resultado una matriz de densidad véase la Figura 14.

Figura 14
Muestra de densidad de características

```
X_vectorizedCV
print(X_vectorizedCV.todense())
[[0 0 1 ... 0 0 0]
 [0 0 0 ... 0 0 0]
 [0 0 1 ... 0 0 0]
 ...
 [0 0 1 ... 0 0 0]
 [0 0 1 ... 0 0 0]
 [0 0 1 ... 0 0 0]]
```

Cuando ya se cuenta con los datos, se tiene que realizar un sobre muestreo de los mismos, lo podemos apreciar en la Figura 15, los cuales al ser diversos pueden extraer de manera completamente desbalanceada los grupos de características de Tipos de evolutivo. Con el sobre muestreo se obtuvo un balance de estos grupos, esto con el fin fundamental de que todos los tipos de evolutivo tengan la misma probabilidad de aparecer sin inclinarse por ningún tipo de evolutivo en específico, para ello se añaden nuevas características clonadas de las previas para emparejar el balance, de esta manera aumenta los 32 elementos del data set, tal como lo podemos apreciar en la Figura 16.

Figura 15
Sobre - muestreo

```
#Importamos los paquetes de sobremuestreo
from imblearn.over_sampling import SMOTE

#SMOTE
smote = SMOTE()
```

Figura 16

Generación de nuevas características

```

--
#Generación de nuevas muestras sintéticas
dataSmoteCV, targetSmoteCV = smote.fit_resample(X_vectorizedCV,target['tipo'])
print(dataSmoteCV)

```

El modelo aprende añadiendo la data de entrenamiento y el resultado, de esta manera se puede realizar una predicción de sobre la misma. Con el método *train_test_split* se separa la data inicial, esto con el fin de realizar grupos de entrenamiento y grupos de evaluación, así el modelo tendrá población con la cuál realizar las pruebas lo podemos apreciar en le Figura 17. El modelo contiene implícitamente un grupo de preguntas con la data añadida, estas podrán, dependiendo de las respuestas, discernir entre uno u otro Tipos de evolutivo, de manera gráfica se puede apreciar el modelo en la Figura 18.

Figura 17

Código del entrenamiento del Árbol de decisión

```

from sklearn.model_selection import train_test_split
dataTrainCV, dataTestCV, targetTrainCV, targetTestCV = train_test_split(dataSmoteCV,targetSmoteCV, random_state = 0)

#Paquete
from sklearn.tree import DecisionTreeClassifier

#Inicialización.
modelDTSCV = DecisionTreeClassifier()

#Entrenamiento
modelDTSCV.fit(dataTrainCV, targetTrainCV)

#Predicción
targetPredDTSCV = modelDTSCV.predict(dataTestCV)

```

Figura 18

Muestra del gráfico del Árbol de decisión

```

#Paquete
from sklearn.tree import plot_tree

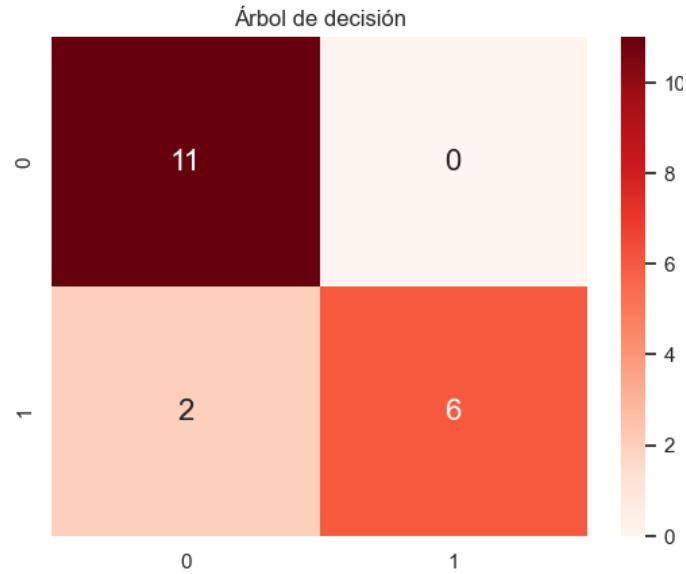
plot_tree(decision_tree=modelDTSCV, feature_names=vectCV.get_feature_names_out(),filled=True, fontsize =8);

```



Los resultados obtenidos con la data de evaluación se grafican en la matriz de confusión del modelo con el fin de medir el mismo y revisar que tan fiable es. Se aprecia, en la Figura 19, 9 celdas correspondientes a falsos positivos, falsos negativos, verdaderos positivos y verdaderos negativos. De los 32 elementos iniciales de la muestra se sobre muestra a 38. 19 elementos son para Test, al generar dentro del algoritmo la mitad de los mismos para prueba.

Figura 19
Matriz de confusión del Árbol de decisión



A continuación, se describe el proceso de implementación Resultados de medición de Machine Learning según los objetivos específicos

V.1. Identificación del tiempo de desarrollo según los puntos de esfuerzo, el tipo de evolutivo, tareas y subtareas del área de desarrollo de software que mejora en la precisión la estimación de tiempos

Para calcular la exactitud, usaremos la fórmula de Accuracy de Machine Learning siguiente:

$$Accuracy = \frac{VP + VN}{VP + FN + FP + FN} = \frac{6 + 11}{6 + 0 + 2 + 11} = 0.8947$$

Para calcular la precisión, usaremos la fórmula de Machine Learning siguiente:

$$Precisión = \frac{VP}{VP + FP} = \frac{6}{6 + 0} = 1$$

Para calcular la exactitud del experto, usaremos la fórmula siguiente:

$$Exactitud\ experto = \frac{valores\ atinados}{valores\ no\ atinados + valores\ atinados} = \frac{26}{32} = 0.8125$$

Tabla 4

Precisión en la estimación de tiempos usando los indicadores de Machine Learning y exactitud del experto

Indicador	Precisión
Exactitud Árbol de decisión	0.894
Exactitud del Experto	0.815
Precisión Árbol de decisión	1

En la tabla 4 se aprecia resultados aceptables en exactitud y precisión.

V.2. Desarrollo de un algoritmo de Machine Learning con un árbol de decisión para detección de tiempos del área de desarrollo de software que mejora en la precisión la estimación de tiempos

Tabla 5
Indicadores del diseño del algoritmo

Indicador	Valor
Confiabilidad	0 fallas internas
Eficiencia	120 segundos
Mantenibilidad	100% de código comentado
Usabilidad	11 pasos para el resultado del modelo

En la tabla 6, En primer lugar, se aprecia ninguna falla interna, lo cual nos indica muy buena respuesta del algoritmo de Python al uso de machine Learning. En segundo lugar, tenemos la eficiencia indicada en 120 segundos como tiempo moderado que depende bastante del computador utilizado pero que para correr el modelo tiene un rendimiento bueno. En tercer lugar, tenemos la mantenibilidad, indicador para el cual se tuvo esfuerzo en comentar todas las líneas de código. Por último, tenemos la usabilidad la cual es de complejidad alta teniendo 11 pasos para su uso, por ello se mejoraría usando un desarrollo de software incluido que interopere en el algoritmo.

V.3. Evaluación de las variables del sistema de estimación de tiempos a tareas y subtareas en el área de desarrollo de software que mejora en la precisión la estimación de tiempos

Para calcular la memoria, usaremos la fórmula de Recall de Machine Learning siguiente:

$$Recall = \frac{VP}{VP + FN} = \frac{6}{6 + 2} = 0.75$$

Para calcular la puntuación del algoritmo, usaremos la fórmula F1 score de Machine Learning siguiente:

$$F1 = 2 \cdot \frac{precision \cdot recall}{precision + recall} = 2 \cdot \frac{0.75 \cdot 1}{0.75 + 1} = 0.8571$$

Tabla 6
Evaluación de las variables

Indicador	Valor
Recall Árbol de decisión	0.75
Puntuación F1 Árbol de decisión	0.8571

En la tabla 7 se puede apreciar las variables de memoria y puntuación F1 score.

Se aprecia en la tabla 7 que los valores positivos correctamente clasificados son el 75% de los casos. De la misma manera se puede apreciar la media armónica de la precisión y los valores positivos correctamente clasificados obteniendo un valor de 0.8571.

De esta manera con los resultados obtenidos se tiene que el modelo mejora la precisión en la estimación de tiempos con respecto a la estimación de tiempos tradicional del 0.8947 en el modelo, el experto tuvo una exactitud de 0.8125.

VI. DISCUSIÓN, CONCLUSIONES Y RECOMENDACIONES

VI.1. Discusión

La investigación planteó como hipótesis que el diseño de un sistema de estimación de tiempos con Machine Learning en el área de Desarrollo de Software tiene una mejor precisión en la estimación de tiempos, esta misma se trianguló en los resultados con la investigación de Villegas Cubas sobre un modelo de machine learning en la detección de sitios web phishing que realiza los mismos cálculos.

Con respecto al resultado del objetivo de la estimación de tiempos según tiempo de desarrollo, tipo de evolutivo, se tuvo una exactitud del 0.8947 en el modelo, el experto tuvo una exactitud de 0.8125, lo cual en comparación con los antecedentes con resultados de 0.8663 como Villegas (2021) la estimación de tiempos. Esta investigación

logra una estimación buena en comparación también con el enfoque tradicional de los tiempos en cuanto a eficacia, los ítems de exactitud fueron satisfactorios para el algoritmo.

Con respecto al resultado del objetivo de desarrollo del algoritmo con Machine Learning, de manera tradicional se realiza una estimación de tiempos manual y requiere la experiencia directa en un proyecto específico y no solo en gestión de proyectos, lo ideal es que el estimador cuente con una herramienta para la estimación, con ello se puede indicar que el diseño del modelo mejora la estimación tradicional de manera operativa y suma experiencia tangible y objetiva de estimaciones pasadas validado con la ISO 9126 se ha obtenido un buen sistema con tiempo moderado, mantenible con todas las líneas de código y con complejidad alta en interoperabilidad. En comparación con los antecedentes, todos mejoran este aspecto siendo el uso de la tecnología un buen aporte para la automatización de la planificación de proyectos de software.

Con respecto resultado del objetivo de la evaluación de variables para mejorar la estimación, se tuvo una memoria del 0.75 y un F1 score del 0.8571 en el modelo y tiene una mejora sustancial con la estimación por ML pero en comparación con los antecedentes no mejoró la estimación directamente como sí lo hizo en los antecedentes obteniendo hasta 0.968 en F1 Score, por tanto no sería necesario realizar una evaluación para mejorar la precisión.

Estos resultados mencionados líneas atrás de estimación de tiempos con las teorías que hacen estos cálculos se puede comparar con los antecedentes de Machine Learning donde utilizan las técnicas para predecir textos cortos como URL de ataque phishing o mensajes de texto de redes sociales obteniendo resultados de 0.935, la diferencia con los textos de tickets de consultoras de software es que los textos son más extensos y por tanto más apegados a error en cuanto al avance de esta investigación se tuvo exactitud

del 0.8947 siendo esta superior a la estimación de tiempo tradicional con lo cual se obtiene el aporte de la misma.

VI.2. Conclusiones

La presente investigación tuvo como objetivo general el desarrollo de un modelo de Machine Learning en el área de Desarrollo de Software que mejora la precisión en la estimación de tiempos. Lo más importante es identificar las librerías correctas de Python para realizar el modelo. Lo que más me ayudó fue la investigación inicial sobre la inteligencia artificial que se muestra a nivel teórico en las librerías y que ayuda en su implementación. Lo más difícil es el desarrollo local dentro del Perú por la cantidad de especialistas que es limitada pero que es una oportunidad para desarrollar los modelos de predicción con Machine Learning. Por consiguiente, para realizar la comparativa humana con la máquina se realizó el modelo cumpliendo con el objetivo general.

Por consecuencia la presente investigación tuvo como objetivo específico la identificación del tiempo de estimación de desarrollo de software según los puntos de esfuerzo, el tipo de evolutivo, tareas y subtareas del área de desarrollo de software para mejorar la precisión en la estimación de tiempos. Lo más importante de identificar el sistema con ML fue la abstracción de las variables y cómo éstas pueden introducirse al modelo porque es la base fundamental del estudio. Lo que más me ayudó a identificar las variables fueron los pasos de estudio para obtener la experiencia en desarrollo porque al indagar sobre tiempos de desarrollo, extraer la mayor cantidad de información sobre ello y luego aplicarlo en el modelo, resulta muy entendible. Lo más difícil al identificar las variables fue la enorme cantidad de composiciones porque se tiene que discernir en diferentes proyectos la identificación de éstas. Este trabajo logró estimar los tiempos de desarrollo según las variables tipo de evolutivo.

De la misma manera se tuvo como objetivo específico el diseño de un modelo con algoritmos de Machine Learning para detección de tiempos del área de desarrollo de software para mejorar la precisión en la estimación de tiempos. Lo más importante del diseño del sistema con ML fue identificar la técnica de ML más adaptable a la investigación porque de eso dependen los resultados. Lo que más me ayudó a diseñar el sistema de detección de tiempos con Machine Learning fue tener accesible la información sobre las técnicas, porque se tuvo el acceso a internet y los dispositivos instalables. Lo más difícil en el diseño del algoritmo fue la poca investigación local sobre el tema, porque es sumamente útil e indispensable tener especialistas investigadores a nivel local. En esta investigación se logró realizar el diseño de un algoritmo con Machine Learning para predecir tiempos de desarrollo de software.

Por último, se tuvo como objetivo específico la evaluación de las variables del sistema de estimación de tiempos a tareas y subtareas en el área de desarrollo de software para mejorar la precisión en la estimación de tiempos. Lo más importante del diseño del sistema con ML fue identificar el lenguaje de programación en el que se realizará el algoritmo porque este puede o no tener las librerías y herramientas necesarias para evaluar las variables y obtener el entrenamiento. Lo que más me ayudó a evaluar las variables del modelo de Machine Learning fue los métodos y librerías disponibles en Python para su desarrollo porque con ello se pudo disminuir considerablemente el tiempo de desarrollo. Lo más difícil al evaluar las variables del algoritmo fue la extracción de la data porque es una elaboración manual de digitar y estructurar la data de acuerdo con cómo se requiere en los modelos de ML.

VI.3. **Recomendaciones**

En esta investigación se concluyó que la estimación de tiempo de desarrollo según los puntos de esfuerzo, el tipo de evolutivo, tareas y subtareas para la extracción de

variables de estimación logró estimar los tiempos de desarrollo según las variables tipo de evolutivo y se recomienda obtener la mayor cantidad de información del proyecto de software a realizar para extraer las variables y poder discernir correctamente las que influyen en el modelo de Machine Learning para estimación de tiempos llevándolo así a futuro un modelo matemático que pueda predecir el comportamiento del Machine Learning sobre la estimación de tiempos, por tanto se recomienda seguir los lineamientos de la documentación libre de Python que ayudan en dichas estimaciones.

De la misma manera con respecto al diseño de un modelo con algoritmos de Machine Learning para detección de tiempos del área de desarrollo de software para mejorar la precisión en la estimación de tiempos, no se logró mejorar al antecedente y se recomienda seguir investigando sobre el tema debido a que existen librerías y lenguajes de programación como Natural Language Toolkit, Spacy, Gensim, los cuales están muy bien documentados que ayudan bastante al investigador a desarrollar los modelos.

En cuanto a la evaluación de las variables del sistema de estimación de tiempos a tareas y subtareas en el área de desarrollo de software para mejorar la precisión en la estimación de tiempos, se logró mejorar la evaluación con respecto al antecedente y se recomienda utilizar Python con Anaconda y Jupyter notebook debido a que tiene un muy buen rendimiento en las computadoras y un alcance basto con librerías que ayudan al desarrollo de modelos con inteligencia artificial, no se recomienda reevaluar las variables en este aspecto debido a que no mejoraría significativamente la precisión.

Lista de referencias

- Almache, M., Raura, G., Ruiz, J., & Fonseca, E. (2021). *Modelo Neuronal de Estimación para el Esfuerzo de Desarrollo en Proyectos de Software (MONEPS)*. Universidad de las Fuerzas Armadas ESPE . <https://doi.org/https://doi.org/10.18294/relais.2015.148-154>
- Araujo Ahon, A., Cárdenas Mayta, B., Iparraguirre Villanueva, O., Zapata Paulini, J., & Cabanillas Carbonel, M. (2023). *Técnicas y algoritmos para predecir el resultado de los partidos de fútbol utilizando la minería de datos*. Universidad Continental.
https://www.researchgate.net/publication/372136077_Tecnicas_y_algoritmos_para_predecir_el_resultado_de_los_partidos_de_futbol_utilizando_la_mineria_de_datos_una_revision_de_la_literatura
- Asenjo, G., Castillo, J., & Muñoz, J. (2017). *Plan de gestión de los procesos alcance, tiempo y costo*. Lima: Universidad Peruana de Ciencias Aplicadas.
- Atlassian. (2023). *Jira Software*.
<https://www.atlassian.com/es/software/jira/guides/issues/overview#what-is-jira-issue-hierarchy>
- Burkov, A. (2019). *The Hundred-Page Machine Learning Book*. Draft.
- Chang, H. M. (2024). *Comparación de técnicas de estimación basadas en machine learning para predecir costos en los planes de adquisiciones de las entidades públicas del Perú*. Universidad Señor de Sipán. <https://hdl.handle.net/20.500.12802/10566>
- Cynoteck. (Diciembre de 2022). *Cynoteck*. <https://cynoteck.com/es/blog-post/software-development-time-estimation/>
- Davila Otani, R. C., & Davila Otani, K. P. (2021). *Desarrollo de una aplicación de análisis del mercado, basado en el procesamiento del lenguaje natural de la red social Twitter, con Machine Learning para predicción de éxito del lanzamiento*. UNIVERSIDAD PERUANA DE CIENCIAS APLICADAS.
https://repositorioacademico.upc.edu.pe/bitstream/handle/10757/657637/Fuentes_DOR.pdf?sequence=3&isAllowed=y
- Drees, F., & Zhang, G. (Agosto de 2021). *www.eluniversal.com*. el universal:
<https://www.eluniversal.com/el-universal/106459/la-importancia-del-acceso-digital>
- Esteban, M., & Pacienza, J. (2015). *Metodologías de desarrollo de software*.
<https://repositorio.uca.edu.ar/bitstream/123456789/522/1/metodologias-desarrollo-software.pdf>
- Geoff Keeling, T. G. (2022). *Enabling Fairness in Healthcare Through Machine Learning*. Springer.
<https://doi.org/10.1007/s10676-022-09658-7>
- Gila Briñas, N. (2021). *Algoritmos avanzados de Machine Learning en la estimación de tiempos de resolución de procesos financieros*. Madrid.
<https://repositorio.comillas.edu/xmlui/handle/11531/54992>

- Hernández Sampieri, R., & Mendoza Torres, C. (2018). *Metodología de la investigación*. McGrawHill.
http://www.biblioteca.cij.gob.mx/Archivos/Materiales_de_consulta/Drogas_de_Abuso/Articulos/SampieriLasRutas.pdf
- Hurwitz, J., & Kirsch, D. (2018). *Machine Learning for dimmies*. IBM Limited Edition.
<https://www.ibm.com/downloads/cas/GB8ZMQZ3>
- Intelequia. (2020). <https://intelequia.com/es/blog/post/ciclo-de-vida-del-software-todo-lo-que-necesitas-saber>. Intelequia.
- Jiménez Campefens, J. N., & Naranjo Ornedo, V. (2020). *Study of depression detection in online social networks using natural language processing and machine learning*. Universidad Politécnica de Valencia. <https://riunet.upv.es/handle/10251/147696>
- LinkedIn. (Junio de 2023). *LinkedIn*. LinkedIn: <https://www.linkedin.com/pulse/am%C3%A9rica-latina-se-destaca-en-el-%C3%A1rea-de-desarrollo-software-hays/?originalSubdomain=es>
- PMO Informática. (2015). *Estimación de proyectos de software por puntos de función*.
<http://www.pmoinformatica.com/2015/04/estimacion-puntos-funcion-introduccion.html>
- Rodríguez Jiménez, A., & Pérez Jacinto, A. O. (2017). *Métodos científicos de indagación y de construcción del conocimiento*. Universidad EAN.
<https://doi.org/https://doi.org/10.21158/01208160.n82.2017.1647>
- Salazar-B, G. (2009). *Estimación de proyectos de software: un caso práctico*. Ingeniería y Ciencia.
<https://www.redalyc.org/articulo.oa?id=83512213006>
- Sanchez Bolivar, J. D., & Cruz Huamán, C. F. (2023). *Simulación de la estimación de tiempos variables de acarreo y movimiento de tierras utilizando el software TALPAC en la etapa de cierre de mina de la Unidad Minera Florencia TUCARI*. Universidad Tecnológica del Perú.
<https://renati.sunedu.gob.pe/handle/sunedu/3421025>
- Schulze, N. (2020). *Desarrollo de software inteligente para la estimación de proyectos*.
<https://oa.upm.es/66676/>
- Vara Horna, A. A. (2015). *Desde la idea hasta la sustentación: 7 pasos para una tesis exitosa*. Universidad de San Martín de Porres. <https://es.slideshare.net/slideshow/7-pasos-para-una-tesis-exitosa/52323061>
- Verity. (2022). *La ISO/IEC 9126: 2001: Características de la calidad de software*.
<https://www.verity.cl/que-es-norma-iso-iec-9126-2001/>
- Villegas Cubas, J. E. (2021). *Modelo de machine learning en la detección de sitios web phishing*. UNIVERSIDAD SEÑOR DE SIPÁN.
<https://repositorio.uss.edu.pe/bitstream/handle/20.500.12802/8897/Villegas%20Cubas%20Juan%20El%20C3%ADas.pdf?sequence=1&isAllowed=y>

Wan Jiang, H., Li Xin, J., Tian Bo, L., & Xiao Yan, Z. (2015). *Comparison of Machine Learning Algorithms for Software Project Time Prediction*. SERSC.
<https://doi.org/10.14257/ijmue.2015.10.9.01>

ZhingFong, L., & Rivas, M. (2023). *Los Factores de Fracaso y Éxito en un proyecto de desarrollo de Software*. Universidad Nacional Mayor de San Marcos.
<https://revistasinvestigacion.unmsm.edu.pe/index.php/rpcsis/article/view/25799/19774>

Apéndice

Anexo I: Matriz de consistencia



MATRIZ DE CONSISTENCIA DE POSGRADO				
AUTOR:	Hugo Jurek Camilo Tumba Burgos			
LÍNEA DE INVESTIGACIÓN:	Tecnologías emergentes			
SUB LÍNEA DE INVESTIGACIÓN:	Machine learning			
TÍTULO*:	Modelo de Machine Learning para la precisión de tiempos del desarrollo de software en una empresa consultora en Lima 2023			
PREGUNTA	OBJETIVOS	HIPÓTESIS	VARIABLES / CATEGORÍA	METODOLOGÍA
1. Pregunta general:	1. Objetivo general:	1. Hipótesis general:	Variable / Categoría 1:	1. Tipo de investigación:
¿De qué manera el desarrollo de un modelo de estimación de tiempos con Machine Learning en el área de Desarrollo de Software mejora la precisión en la estimación de tiempos?	Desarrollar un modelo de detección de tiempos con Machine Learning en el área de Desarrollo de Software que mejora la precisión en la estimación de tiempos.	El desarrollo de un modelo de estimación de tiempos con Machine Learning en el área de Desarrollo de Software tiene una mejor precisión en la estimación de tiempos.	Modelo de estimación de tiempos con Machine Learning en el área de Desarrollo de Software	Cuantitativa
2. Preguntas específicas:	2. Objetivos específicos:	2. Hipótesis específicas:	Variable / Categoría 2:	2. Nivel de investigación:
P1: ¿De qué manera identificar el tiempo de desarrollo según los puntos de esfuerzo, el tipo de evolutivo, tareas y subtareas del área de desarrollo de software mejora la precisión en la estimación de tiempos?	O1: Identificar el tiempo de desarrollo según los puntos de esfuerzo, el tipo de evolutivo, tareas y subtareas del área de desarrollo de software que mejora en la precisión la estimación de tiempos.	H1: El tiempo de desarrollo según los puntos de esfuerzo, el tipo de evolutivo, tareas y subtareas del área de desarrollo de software tiene una mejor precisión en la estimación de tiempos.	Precisión en la estimación de tiempos.	Aplicada
P2: ¿De qué manera el desarrollo de un algoritmo de machine Learning por refuerzo para detección de tiempos del área de desarrollo de software mejora la precisión en la estimación de tiempos?	O2: Desarrollar un algoritmo de machine Learning con un árbol de decisión para detección de tiempos del área de desarrollo de software que mejora en la precisión la estimación de tiempos.	H2: El algoritmo de machine Learning para estimación de tiempos del área de desarrollo de software tiene una mejor precisión en la estimación de tiempos.		3. Diseño de la investigación:
P3: ¿De qué manera dimensionar las variables del sistema de estimación de tiempos a 2 tareas y subtareas en el área de desarrollo de software mejora la precisión en la estimación de tiempos?	O3: Evaluar las variables del sistema de estimación de tiempos a tareas y subtareas en el área de desarrollo de software que mejora la precisión en la estimación de tiempos.	H3: Las variables del modelo de estimación de tiempos a tareas y subtareas en el área de desarrollo de software tiene una mejor precisión en la estimación de tiempos.		experimental
				4. Método:
				Analítico sintético
				5. Población:
				Data set en jira

Anexo II

Instrumento: Ficha de guía de observación. Medición de variable independiente.

Para medir el desarrollo del algoritmo, que son en sí mismas líneas de código dentro de un sistema con el lenguaje de programación Python. Para ello se utiliza la ISO 9126.

Dimensiones	Indicadores	Escala de medición		
Desarrollo del algoritmo	Confiabilidad	1 a 2	3 a 5	más de 5
	Eficiencia	Cantidad en segundos		
	Mantenibilidad	número porcentual		
	Usabilidad	1 a 5	5 a 10	10 a más

Cada indicador tiene su respectiva fórmula indicada a continuación:

$$\text{Confiabilidad} = \text{Número de fallas internas}$$

$$\text{Eficiencia} = \text{Tiempo de uso (segundos)}$$

$$\text{Mantenibilidad} = \frac{\text{Cantidad de código procedimental comentado}}{\text{Cantidad total de código procedimental}}$$

$$\text{Usabilidad} = \text{Cantidad de pasos para el uso}$$

Anexo III

Instrumento: Ficha de guía de observación. Medición de variable dependiente.

Dimensione s	Indicadores	Escala de medición
Precisión en la estimación de tiempos	Precisión	0-1
	Accuracy	0-1
Evaluación de las variables	Recall	0-1
	F1-Score	0-1

Cada indicador tiene su respectiva fórmula indicada a continuación:

$$Accuracy = \frac{VP+VN}{VP+FN+FP+FN} \quad (4)$$

$$Precisión = \frac{VP}{VP+FP} \quad (5)$$

$$Recall = \frac{VP}{VP+FN} \quad (6)$$

$$F1 = 2 \cdot \frac{precision \cdot recall}{precision + recall} \quad (7)$$

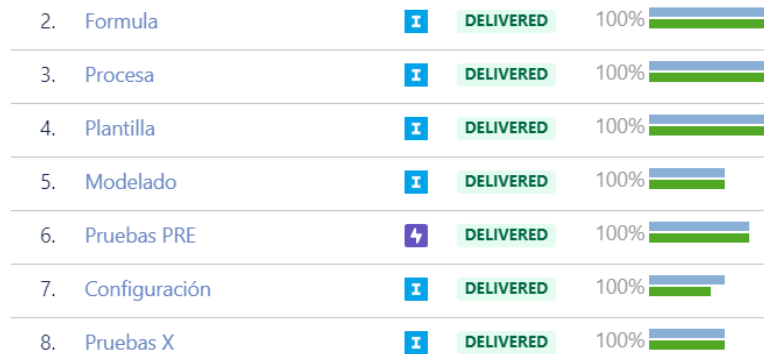
Anexo IV

Información de tiempos de desarrollo en jira.

En global

SUB-TIPO	OBSERVACIONES NAOS	REDMINE	ESTADO	ASUNTO	CREADO	OBSERVACIONES HISTORICO	
0	CEXVEAIA-5380	23/06/2023 ENTREGADO - En el Anexo I, al tener dos provincias distintas entre edificio y el ILMO, lanza una validación y no permite que avance a	1378455	261830	P. Cerrada	[CFATV] [24794] [OTR#1378455] Problemas pre-carga de anexo II	15/05/2023 13:15
2	CEXVEAIA-5376	23/06/2023 Alberto - 21/07/2023 - Mantenemos reunión con Ramón y el equipo de Fomento e indicamos que realicen pruebas en PXP al estar la m	1334372	261868	P. Bloqueada	[CFATV] [24808] [XXX#1334372] Defecto incidencia grabado solicit	30/05/2023
0	CEXVEAIA-5284	15/06/2023 ENTREGADO	1388497	262038	P. Cerrada	[CALFFP] [XXX#1388497] HABIRIA: ACU En el El formulario no sale	27/06/2023 18:30
0		5/07/2023 Se ha resuelto por parte de Solutions. No entendemos que acción se ha realizado, pero el mensaje del log era claro: no encuentra la fi	1343022	262039	P. Cerrada	[CALFFP] [XXX#1343022] Error al realizar presentación FHCN Agru	27/06/2023 18:45
0	CEXVEAIA-5289	ds REALIZADO, el detalle de todo lo que se ha detectado se encuentra en el word, adjunto en la tarea JIRA CEXVEAIA-5289 (incidencias - N	1388347	262043	Entregada	[CALFFP] [DEF#1388347] HABIRIA: Nombramiento provisional (NPV)	18/06/2023 07:00
0	CEXVEAIA-5383	13/07/2023 13-07-2023(Hugo): Realizado - 6 Procedimientos PNCN y 2 anexos.	1391843	262074	Entregada	[CALFFP] [DEF#1391843] Revisión de los procedimientos de Modif	13/06/2023 14:45
0	CEXVEAIA-5390	7/07/2023 12-07-2023(Hugo): Realizado - se actualiza el UI del texto alternativo en CCP. En la incidencia se indica que en la fase de Subsanac	1392968	262088	Entregada	[CALFFP] [DEF#1392968] HABIRIA: Procedimiento CCP.No hay botton	14/06/2023 16:30
0	CEXVEAIA-5386	27/06/2023 ENTREGADO: Se realiza una actualización en Anexo I y Anexo II Carga Inicial.	1394418	262113	P. Cerrada	[CFATV] [24793] [OTR#1394418] Quitar "" de campos de apellido	19/06/2023 08:43
0	CEXVEAIA-5386	27/06/2023 ENTREGADO: Se realiza una actualización en Anexo I y Anexo II Carga Inicial.	1395847	262122	P. Cerrada	[CFATV] [24793] [OTR#1395847] Quitar "" de campos de apellido	19/06/2023 14:30
0	CEXVEAIA-5386	27/06/2023 ENTREGADO: Se realiza una actualización en Anexo I y Anexo II Carga Inicial.	1396148	262127	P. Cerrada	[CFATV] [24793] [XXX#1396148] Quitar "" de campos de apellido	20/06/2023 09:15
0	CEXVEAIA-5372	27/06/2023 ENTREGADO - Se realiza los ajustes en ambos formularios, PFP y pruebas	1397749	262140	P. Cerrada	[CEHFE] [XXX#1397749] VEAIA PXP: modificación formularios Emer	21/06/2023 12:35
0	CEXVEAIA-5386	27/06/2023 ENTREGADO: Se realiza una actualización en Anexo I y Anexo II Carga Inicial.	1394738	262141	P. Cerrada	[CFATV] [24793] [XXX#1394738] Quitar "" de campos de apellido	21/06/2023 12:35
1		13/07/2023 he puesto comentario en naos para que comprueben de nuevo los procedimientos porque estaban seleccionando el de otro año y con	1391070	262162	P. Cerrada	[CSC] [22629, 22631] [OTR#1391070] Expedientes relacionados de	22/06/2023 09:15
0	CEXVEAIA-5374	27/06/2023 ENTREGADO	1398992	262163	P. Verificada	[CEHFE] [XXX#1398992] VEAIA PXP: Modificación en formulario de	22/06/2023 11:45

A detalle:



Anexo V

Documento de Instrumentos presentados a los validadores por juicio de experto.

DEFINICIÓN CONCEPTUAL DE LAS VARIABLES Y DIMENSIONES

Variable 1: Sistema de detección de tiempos con Machine Learning en el área de Desarrollo de Software

Según Burkov (2019) Machine Learning es una rama de la informática que se centra en el desarrollo de algoritmos que, para ser efectivos, se apoyan en un conjunto de ejemplos de un fenómeno dado. Estos ejemplos pueden originarse en la naturaleza, ser creados manualmente por personas o generados por otros algoritmos.

Dimensiones de las variables:

Dimensión: Diseño del algoritmo.

En esta dimensión se medirá la confiabilidad, eficiencia, mantenibilidad y usabilidad del algoritmo según la ISO 9126.

MATRIZ DE OPERACIONALIZACIÓN DE LAS VARIABLES

Variable 1: Sistema de detección de tiempos con Machine Learning en el área de Desarrollo de Software.

Variable	Dimensiones	indicadores	items	Escala	Niveles o rangos
Sistema de detección de tiempos con Machine Learning en el área de Desarrollo de Software	Diseño del algoritmo	Confiabilidad	Cantidad de caídas	Unidad	Razón
		Eficiencia	Tiempo de ejecución de una predicción	Cantidad en segundos	Razón

		Mantenibilidad	Porcentaje de código comentado	Porcentaje	Porcentaje
		Usabilidad	Cantidad de pasos para el uso	Unidad	Razon



Cada indicador tiene su respectiva fórmula indicada a continuación:

Confiabilidad = Número de fallas internas

Eficiencia = Tiempo de uso (segundos)

Mantenibilidad = $\frac{\text{Cantidad de código procedimental comentado}}{\text{Cantidad total de código procedimental}}$

Usabilidad = Cantidad de pasos para el uso

INTRUMENTO DE EVALUACIÓN

Instrumento: Ficha de guía de observación. Medición de variable independiente.

Para medir el diseño del algoritmo, que son en sí mismas líneas de código dentro de un sistema con el lenguaje de programación Python. Para ello se utiliza la ISO 9126.

Dimensiones	Indicadores	Escala de medición		
Diseño del algoritmo	Confiabilidad	1 a 2	3 a 5	más de 5
	Eficiencia	Cantidad en segundos		
	Mantenibilidad	número porcentual		
	Usabilidad	1 a 5	5 a 10	10 a más

Cada indicador tiene su respectiva fórmula indicada a continuación:

Confiabilidad = Número de fallas internas

Eficiencia = Tiempo de uso (segundos)

Mantenibilidad = $\frac{\text{Cantidad de código procedimental comentado}}{\text{Cantidad total de código procedimental}}$

Usabilidad = Cantidad de pasos para el uso

DEFINICIÓN CONCEPTUAL DE LAS VARIABLES Y DIMENSIONES

Variable 2: Precisión en la estimación de tiempos

Según Salazar-B (2009) precisión en la estimación del tiempo de desarrollo de software implica la aproximación de un periodo calculado para la finalización de todas las tareas. Estas tareas abarcan diversos conjuntos de actividades necesarias para llevar a cabo el desarrollo de un producto. La estimación global de la tarea varía en función de diversos factores, como la complejidad del proyecto, la cantidad de mano de obra necesaria, la experiencia general del equipo y la disponibilidad de recursos.

Dimensiones de las variables:

Dimensión 1: Precisión en la estimación de tiempos.

En esta dimensión se medirá la precisión y exactitud del modelo según las métricas internacionales conocidas con datos extraídos de la matriz de confusión.

Dimensión 2: Evaluación de las variables

En esta dimensión se medirá la memoria y el modelo según las métricas internacionales conocidas con datos extraídos de la matriz de confusión. Recall y F1 Score.

Variable 2: Precisión en la estimación de tiempos.

Variable	Dimensiones	indicadores	items	Escala	Niveles o rangos
Sistema de detección de tiempos con Machine Learning en el área de Desarrollo de Software	Evaluación de las variables	Precisión	Fórmula precisión	Fórmula precisión	Razón
		Accuracy	Fórmula Accuracy	Fórmula Accuracy	Razón

		Recall	Fórmula Recall	Fórmula Recall	Razón
	Precisión en la estimación de tiempos	F1-Score	Fórmula F1	Fórmula F1	Razón

Cada indicador tiene su respectiva fórmula indicada a continuación:

$$Accuracy = \frac{VP+VN}{VP+FN+FP+FN} \quad (4)$$

$$Precisión = \frac{VP}{VP+FP} \quad (5)$$

$$Recall = \frac{VP}{VP+FN} \quad (6)$$

$$F1 = 2 \cdot \frac{precision \cdot recall}{precision + recall} \quad (7)$$

INTRUMENTO DE EVALUACIÓN

Instrumento: Ficha de guía de observación. Medición de variable dependiente.

Dimensiones	Indicadores	Escala de medición
Precisión en la estimación de tiempos	Precisión	0-1
	Accuracy	0-1
Evaluación de las variables	Recall	0-1
	F1-Score	0-1

Cada indicador tiene su respectiva fórmula indicada a continuación:

$$Accuracy = \frac{VP+VN}{VP+FN+FP+FN} \quad (4)$$

$$Precisión = \frac{VP}{VP+FP} \quad (5)$$

$$Recall = \frac{VP}{VP+FN} \quad (6)$$

$$F1 = 2 \cdot \frac{precision \cdot recall}{precision + recall} \quad (7)$$

Anexo VI

Validación de instrumentos por juicio de expertos: Dr. Edmundo Rafael Casavilca Maldonado
Doctor en Contabilidad y Finanzas, Magister en Administración de la Universidad de Québec
en Montreal, Canadá.

CARTA DE PRESENTACIÓN

Señor Dr.: Casavilca Maldonado Edmundo Rafael

Presente

Asunto: Validación de instrumentos a través de juicio de experto

Me es muy grato comunicarme con usted para expresarle mi saludo y, asimismo, hacer de su conocimiento que, como estudiante del programa de Maestría en Ingeniería de Sistemas de la Universidad Privada del Norte, requiero validar el instrumento con el cual recogeré la información necesaria para poder desarrollar mi trabajo de investigación.

El título de mi investigación es "Diseño de un sistema de estimación de tiempos con Machine Learning en el área de Desarrollo de Software en una empresa de consultoría en Lima en el año 2023" y al ser imprescindible contar con la aprobación de docentes especializados para poder aplicar los instrumentos en mención, he considerado conveniente recurrir a usted, ante su connotada experiencia en temas educativos y/o investigación educativa.

El expediente de validación, que le hago llegar contiene:

- Carta de presentación.
- Definiciones conceptuales de las variables y dimensiones.
- Matriz de operacionalización de las variables.

Le expreso mis sentimientos de respeto y consideración. Me despido de usted, no sin antes agradecerle por la atención a la presente.

Atentamente,

.....
Hugo Jurek Camilo Tumba Burgos
DNI: 48334163

CERTIFICADO DE VALIDEZ DE CONTENIDO DEL INSTRUMENTO QUE MIDE LA VARIABLE I: Sistema de detección de tiempos con Machine Learning en el área de Desarrollo de Software

N°	DIMENSIONES / ítems	Pertinencia ¹		Relevancia ²		Claridad ³		Sugerencias
		Si	No	Si	No	Si	No	
	DIMENSIÓN 1: Diseño del algoritmo							
1	Confiabilidad	X		x		x		
2	Eficiencia	X		x		x		
3	Mantenibilidad	X		x		x		
4	Usabilidad	x		x		x		

Observaciones (precisar si hay suficiencia):

Opinión de aplicabilidad: Aplicable [x] Aplicable después de corregir [] No aplicable []

Apellidos y nombres del juez validador: Dr: Edmundo Rafael Casavilca Maldonado DNI: 06598217

Especialidad del validador: Doctor en Contabilidad y Finanzas

Lima, 25 de Noviembre de 2023



.....
Firma del experto informante

¹Pertinencia: El ítem corresponde al concepto teórico formalizado.

²Relevancia: El ítem es apropiado para representar al componente o dimensión específica del constructo.

³Claridad: Se entiende sin dificultad alguna el enunciado del ítem, es conciso, exacto y directo.

Nota: Se dice suficiencia cuando los ítems planteados son suficientes para medir la dimensión.

CERTIFICADO DE VALIDEZ DE CONTENIDO DEL INSTRUMENTO QUE MIDE LA VARIABLE 2 Precisión en la estimación de tiempos

N°	DIMENSIONES / ítems	Pertinencia ¹		Relevancia ²		Claridad ³		Sugerencias
		Sí	No	Sí	No	Sí	No	
	DIMENSIÓN 1: Evaluación de las variables							
1	Precisión	X		X		X		
2	Accuracy	X		X		X		
	DIMENSIÓN 2: Precisión en la estimación de tiempos							
6	Recall	X		X		X		
7	F1-Score	X		X		X		

Observaciones (precisar si hay suficiencia):

Opinión de aplicabilidad: Aplicable [X] Aplicable después de corregir [] No aplicable []

Apellidos y nombres del juez validador: Dr Edmundo Rafael Casavilca Maldonado DNI:06598217

Especialidad del validador: Doctor en Contabilidad y Finanzas

Lima, 25 de Noviembre de 2023



Firma del experto informante

¹Pertinencia: El ítem corresponde al concepto teórico formulado.

²Relevancia: El ítem es apropiado para representar al componente o dimensión específica del constructo.

³Claridad: Se entiende sin dificultad alguna el enunciado del ítem, es conciso, exacto y directo.

Note: Se dice suficiencia cuando los ítems planteados son suficientes para medir la dimensión.

Anexo VII

Validación de instrumentos por juicio de expertos: Mg. Alfredo Mori Tello

Especialista en Inteligencia artificial, con Maestría en Ingeniería de Sistemas.

CARTA DE PRESENTACIÓN

Señor Mg.: Alfredo Mori Tello

Presente

Asunto: Validación de instrumentos a través de juicio de experto

Me es muy grato comunicarme con usted para expresarle mi saludo y, asimismo, hacer de su conocimiento que, como estudiante del programa de Maestría en Ingeniería de Sistemas de la Universidad Privada del Norte, requiero validar el instrumento con el cual recogeré la información necesaria para poder desarrollar mi trabajo de investigación.

El título de mi investigación es "Diseño de un sistema de estimación de tiempos con Machine Learning en el área de Desarrollo de Software en una empresa de consultoría en Lima en el año 2023" y al ser imprescindible contar con la aprobación de docentes especializados para poder aplicar los instrumentos en mención, he considerado conveniente recurrir a usted, ante su connotada experiencia en temas educativos y/o investigación educativa.

El expediente de validación, que le hago llegar contiene:

- Carta de presentación.
- Definiciones conceptuales de las variables y dimensiones.
- Matriz de operacionalización de las variables.

Le expreso mis sentimientos de respeto y consideración. Me despido de usted, no sin antes agradecerle por la atención a la presente.

Atentamente,



.....
Hugo Jurek Camilo Tumba Burgos
DNI: 48334163

CERTIFICADO DE VALIDEZ DE CONTENIDO DEL INSTRUMENTO QUE MIDE LA VARIABLE 2 Precisión en la estimación de tiempos

N°	DIMENSIONES / ítems	Pertinencia ¹		Relevancia ²		Claridad ³		Sugerencias
		Si	No	Si	No	Si	No	
	DIMENSIÓN 1: Evaluación de las variables							
1	Precisión	X		X		X		
2	Accuracy	X		X		X		
	DIMENSIÓN 2: Precisión en la estimación de tiempos							
6	Recall	X		X		X		
7	F1-Score	X		X		X		

Observaciones (precisar si hay suficiencia):

Opinión de aplicabilidad: Aplicable [X] Aplicable después de corregir [] No aplicable []

Apellidos y nombres del juez validador: Dr Alfredo Mori Tello

DNI: 10721115

Especialidad del validador: Doctor en Ingeniería de Sistemas

Lima, 26 de Octubre de 2023



Firma del experto Informante

¹**Pertinencia:** El ítem corresponde al concepto teórico formulado.

²**Relevancia:** El ítem es apropiado para representar al componente o dimensión específica del constructo.

³**Claridad:** Se entiende sin dificultad alguna el enunciado del ítem, es conciso, exacto y directo.

Nota: Se dice suficiencia cuando los ítems planteados son suficientes para medir la dimensión.

Anexo VIII

Validación de instrumentos por juicio de expertos: Mg. Kenny Mayer Caballero Rodríguez

Especialista en Ciencia de Datos, Estadístico, con diplomados en Inteligencia artificial.

CARTA DE PRESENTACIÓN

Señor Mg: Kenny Mayer Caballero Rodríguez

Presente

Asunto: Validación de instrumentos a través de juicio de experto

Me es muy grato comunicarme con usted para expresarle mi saludo y, asimismo, hacer de su conocimiento que, como estudiante del programa de Maestría en Ingeniería de Sistemas de la Universidad Privada del Norte, requiero validar el instrumento con el cual recogeré la información necesaria para poder desarrollar mi trabajo de investigación.

El título de mi investigación es "Diseño de un sistema de estimación de tiempos con Machine Learning en el área de Desarrollo de Software en una empresa de consultoría en Lima en el año 2023" y al ser imprescindible contar con la aprobación de docentes especializados para poder aplicar los instrumentos en mención, he considerado conveniente recurrir a usted, ante su connotada experiencia en temas educativos y/o investigación educativa.

El expediente de validación, que le hago llegar contiene:

- Carta de presentación.
- Definiciones conceptuales de las variables y dimensiones.
- Matriz de operacionalización de las variables.

Le expreso mis sentimientos de respeto y consideración. Me despido de usted, no sin antes agradecerle por la atención a la presente.

Atentamente,



.....
Hugo Jurek Camilo Tumba Burgos
DNI: 48334163

CERTIFICADO DE VALIDEZ DE CONTENIDO DEL INSTRUMENTO QUE MIDE LA VARIABLE 1: Sistema de detección de tiempos con Machine Learning en el área de Desarrollo de Software

N°	DIMENSIONES / ítems	Pertinencia ¹		Relevancia ²		Claridad ³		Sugerencias
		Si	No	Si	No	Si	No	
	DIMENSIÓN 1: Diseño del algoritmo							
1	Confiabilidad	X		X		X		
2	Eficiencia	X		X		X		
3	Mantenibilidad	X		X		X		
4	Usabilidad	X		X		X		

Observaciones (precisar si hay suficiencia):

Opinión de aplicabilidad: Aplicable [x] Aplicable después de corregir [] No aplicable []

Apellidos y nombres del juez validador: Mg: Caballero Rodríguez Kenny Mayer DNI: 71468773

Especialidad del validador: Maestría en Ciencia de datos

Lima, 20 de Octubre de 2023

Firma del experto informante

¹**Pertinencia:** El ítem corresponde al concepto teórico formulado.

²**Relevancia:** El ítem es apropiado para representar al componente o dimensión específica del constructo.

³**Claridad:** Se entiende sin dificultad alguna el enunciado del ítem, es conciso, exacto y directo.

Note: Se dice suficiencia cuando los ítems planteados son suficientes para medir la dimensión.

CERTIFICADO DE VALIDEZ DE CONTENIDO DEL INSTRUMENTO QUE MIDE LA VARIABLE 2 Precisión en la estimación de tiempos

N°	DIMENSIONES / ítems	Pertinencia ¹		Relevancia ²		Claridad ³		Sugerencias
		Si	No	Si	No	Si	No	
	DIMENSIÓN 1: Evaluación de las variables							
1	Precisión	X		X		X		
2	Accuracy	X		X		X		
	DIMENSIÓN 2: Precisión en la estimación de tiempos							
6	Recall	X		X		X		
7	F1-Score	X		X		X		

Observaciones (precisar si hay suficiencia):

Opinión de aplicabilidad: Aplicable [X] Aplicable después de corregir [] No aplicable []

Apellidos y nombres del juez validador: Mg: Caballero Rodríguez Kenny Mayer DNI: 71468773

Especialidad del validador: Maestría en Ciencia de datos

Lima, 20 de Octubre de 2023

Firma del experto Informante

¹**Pertinencia:** El ítem corresponde al concepto teórico formulado.

²**Relevancia:** El ítem es apropiado para representar al componente o dimensión específica del constructo.

³**Claridad:** Se entiende sin dificultad alguna el enunciado del ítem, es conciso, exacto y directo.

Nota: Se dice suficiencia cuando los ítems planteados son suficientes para medir la dimensión.