UNIVERSIDAD NACIONAL DEL SANTA

ESCUELA DE POSGRADO

PROGRAMA DE DOCTORADO EN INGENIERÍA DE SISTEMAS E INFORMATICA

"Modelo predictivo basado en Machine Learning Supervised y la deserción estudiantil en centros de Educación Superior Tecnológicos Públicos de la región la Libertad"

Tesis para Obtener el Grado de Doctor en Ingeniería de Sistemas e Informática

Autor:

Mg. Polo Romero, Víctor Jaime

Asesor:

Dr. Gutiérrez Gutiérrez, Jorge Luis
DNI Nº. 18135227
Código ORCID: 0000-0002-4989-1196

Nuevo Chimbote - PERÚ
2024

CONSTANCIA DE ASESORAMIENTO DE TESIS

Yo, Dr. Gutierrez Gutierrez, Jorge Luis, a través del presente documento, hago constar mi asesoramiento de la Tesis de Doctorado titulado: **Modelo predictivo basado en Machine Learning Supervised y la deserción estudiantil en centro de Educación Tecnológicos Públicos de la Región la Libertad,** que tiene como autor al: **Mg. Polo Romero, Víctor Jaime,** alumno del Doctorado en Ingeniería de Sistemas e Informática, ha sido elaborado de acuerdo al Reglamento de Normas y Procedimientos para optar el Grado de **Doctor** de la Escuela de Posgrado de la Universidad Nacionaldel Santa;

Dr. Gutierrez Gutierrez, Jorge Luis
ASESOR
DNI: 18135227
Código ORCID: 0000-0002-4989-1196

UNS
ESCUELA DE POSGRADO

AVAL DE CONFORMIDAD DEL JURADO

Tesis de Doctorado titulado: **Modelo predictivo basado en Machine Learning Supervised y la deserción estudiantil en centro de Educación Tecnológicos Públicos de la Región la Libertad,** que tiene como autor al: **Mg. Polo Romero, Víctor Jaime,** alumno del Doctorado en Ingeniería de Sistemas e Informática.

Revisado y Aprobado por el Jurado Evaluador:

Dr. Guerra Cordero, Carlos
Presidenta
DNI: 32739372
ORCID: 0000-0001-6096-4010

Dra. Briones Pereyra, Lisbeth Dora
Secretaria
DNI: 32960646
ORCID 0000-0003-0623-7227

Dr. Gutierrez Gutierrez, Jorge Luis
ASESOR
DNI: 18135227
ORCID: 0000-0002-4989-1196

UNS
ESCUELA DE POSGRADO

ACTA DE EVALUACIÓN DE SUSTENTACIÓN DE TESIS

A los cinco días del mes de julio del año 2024, siendo las 10:00 horas, en el aula P-01 de la Escuela de Posgrado de la Universidad Nacional del Santa, se reunieron los miembros del Jurado Evaluador, designados mediante Resolución Directoral Nº 090-2024-EPG-UNS de fecha 29.02.2024, conformado por los docentes: Dr. Carlos Guerra Cordero (Presidente), Dra. Lizbeth Dora Briones Pereyra (Secretaria) y Dr. Jorge Luis Gutiérrez Gutiérrez (Vocal); con la finalidad de evaluar la tesis titulada ***MODELO PREDICTIVO BASADO EN MACHINE LEARNING SUPERVISED Y LA DESERCÓN ESTUDIANTIL EN CENTRO DE EDUCACIÓN SUPERIOR TECNOLÓGICOS PÚBLICOS DE LA REGIÓN LA LIBERTAD;*** presentado por el tesista **Víctor Jaime Polo Romero,** egresado del programa de **Doctorado en Ingeniería de Sistemas e Informática.**

Sustentación autorizada mediante Resolución Directoral N° 330-2024-EPG-UNS de fecha 25 de junio de 2024.

El presidente del jurado autorizó el inicio del acto académico; producido y concluido el acto de sustentación de tesis, los miembros del jurado procedieron a la evaluación respectiva, haciendo una serie de preguntas y recomendaciones al tesista, quien dio respuestas a las interrogantes y observaciones.

El jurado después de deliberar sobre aspectos relacionados con el trabajo, contenido y sustentación del mismo y con las sugerencias pertinentes, declara la sustentación como Aprobado, asignándole la calificación de 19.

Siendo las 11:00 horas del mismo día se da por finalizado el acto académico, firmando la presente acta en señal de conformidad.

Dr. Carlos Guerra Cordero
Presidente

Dra. Lizbeth Dora Briones Pereyra
Secretaria

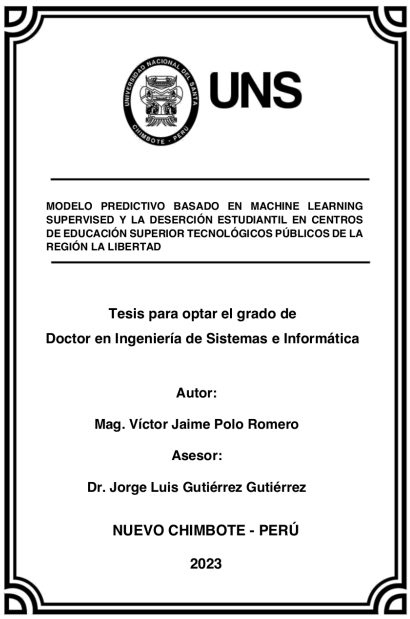
Dr. Jorge Luis Gutiérrez Gutiérrez
Vocal

Recibo digital

Este recibo confirma quesu trabajo ha sido recibido por Turnitin. A continuación podrá ver la información del recibo con respecto a su entrega.

La primera página de tus entregas se muestra abajo.

Autor de la entrega:	Victor Jaime POLO ROMERO
Título del ejercicio:	DOCTORADO 2023
Título de la entrega:	MODELO PREDICTIVO BASADO EN MACHINE LEARNING SUP...
Nombre del archivo:	TesisFinal_JaimePoloRomero30.pdf
Tamaño del archivo:	1.19M
Total páginas:	110
Total de palabras:	22,562
Total de caracteres:	124,439
Fecha de entrega:	30-ene.-2024 07:16a. m. (UTC-0500)
Identificador de la entre...	2140657588



UNS

MODELO PREDICTIVO BASADO EN MACHINE LEARNING SUPERVISED Y LA DESERCIÓN ESTUDIANTIL EN CENTROS DE EDUCACIÓN SUPERIOR TECNOLÓGICOS PÚBLICOS DE LA REGIÓN LA LIBERTAD

Tesis para optar el grado de

Doctor en Ingeniería de Sistemas e Informática

Autor:

Mag. Víctor Jaime Polo Romero

Asesor:

Dr. Jorge Luis Gutiérrez Gutiérrez

NUEVO CHIMBOTE - PERÚ

2023

MODELO PREDICTIVO BASADO EN MACHINE LEARNING SUPERVISED Y LA DESERCIÓN ESTUDIANTIL EN CENTROS DE EDUCACIÓN SUPERIOR TECNOLÓGICOS PÚBLICOS DE LA REGIÓN LA LIBERTAD

INFORME DE ORIGINALIDAD

22%	22%	2%	11%
INDICE DE SIMILITUD	FUENTES DE INTERNET	PUBLICACIONES	TRABAJOS DEL ESTUDIANTE

FUENTES PRIMARIAS

1	repositorio.uns.edu.pe Fuente de Internet	3%
2	hdl.handle.net Fuente de Internet	1%
3	aws.amazon.com Fuente de Internet	1%
4	repositorio.unam.edu.pe Fuente de Internet	1%
5	aprendeia.com Fuente de Internet	1%
6	docplayer.es Fuente de Internet	1%
7	repositorio.utc.edu.ec Fuente de Internet	1%
8	repositorio.utp.edu.pe Fuente de Internet	1%

ÍNDICE DE CONTENIDO

CERTIFICACIÓN DEL ASESOR……. ii
AVAL DEL JURADO……...iii
RESUMEN…….vii
ABSTRACT……viii
INTRODUCCION.....X
CAPÍTULO I.- Problema de Investigación 12
1.1. Planteamiento y fundamentación del problema de investigación 12
1.2. Antecedentes de la investigación 14
1.2.1. Internacionales 14
1.2.2. Nacionales 15
1.2.3. Locales 18
1.3. Formulación del problema de investigación 18
1.4. Delimitación del estudio 18
1.5. Justificación e importancia de la investigación 19
1.6. Objetivos de la investigación 21
CAPÍTULO II.- Marco Teórico 22
2.1. Fundamentos teóricos de la investigación 22
2.2. Marco conceptual 58
CAPÍTULO III.- Marco Metodológico 62
3.1. Hipótesis central de la investigación 62
3.2. Variables e indicadores de la investigación 62
3.3. Método de la Investigación 63
3.4. Diseño 64
3.5. Población y Muestra 66
3.6. Técnicas e Instrumentos de Recolección de Datos 66
3.7. Procedimiento de la Recolección de Datos 66
3.8. Técnicas de Procesamiento y análisis de Resultados 67
3.9. Colección de datos caso estudio 67

CAPÍTULO IV.- Resultados y Discusión 69

RESULTADOS 69

DISCUSION 101

CAPÍTULO V.-Conclusiones y Recomendaciones 104

CONCLUSIONES 104

RECOMENDACIONES 105

REFERENCIAS BIBLIOGRÁFICAS 106

ANEXO 1 Código Programa 110

ANEXO 2 Predicción con data real al azar 137

ANEXO 3 Juicio de expertos 138

ÍNDICE DE TABLAS

Tabla 1 *Lista de variables planteadas por Bean*.....37

Tabla 2 Técnicas de predicción utilizadas (1971-2008) 39

Tabla 3 Comparación KDD - SEMMA 55

Tabla 4 Comparación KDD – CRISP - DM 56

Tabla 5 Comparación SEMA – CRISP – DM 56

Tabla 6 La distribución regional de los votantes 57

Tabla 7 Indicadores.....63

Tabla 8 Caracteristica identificadas inicialmente para el Data Set.....74

Tablas 9 Tabla de correlación Pearson…….....75

ÍNDICE DE FIGURAS

Figura 1 Ciclo de vida de un modelo predictivo 24
Figura 2 Sistema típico de Voting 31
Figura 3 Esquema del proceso Bagging 31
Figura 4 Esquema del proceso Boosting 32
Figura 5 Niveles y tipos de deserción 34
Figura 6 Modelo predictivo planteado por Spady 35
Figura 7 Modelo planteado por Tinto 36
Figura 8 Categorías o variables que inciden en la persistencia o abandono estudiantil 38
Figura 9 Función lineal 43
Figura 10 Ecuación de regresión logística 43
Figura 11 Gráfica de la regresión logística 44
Figura 12 Ecuación de la regresión logística múltiple 44
Figura 13 Representación de la función logística 45
Figura 14 Estructura de un árbol de decisión 46
Figura 15 Ejemplo de un árbol de decisión 46
Figura 16 Regresión con bosques aleatorios 47
Figura 17 Estructura de un Bosque aleatorios 48
Figura 18 Ejemplo de un Bosque aleatorio 48
Figura 19 Hiperplanos generados 49
Figura 20 H3 como mejor recta de clasificación 50
Figura 21 Representa una regresión con Naive Bayes 50
Figura 22 Etapas del proceso KDD 53
Figura 23 Fases y Actividades del Procesos SEMMA 53
Figura 24 Fases de la metodología CRISP 54
Figura 25 Diseño de la investigación 65
Figura 26 Metodología CRISP DM 69
Figura 27 Comprensión del negocio 70
Figura 28 Comprensión del negocio 71
Figura 29 Comprensión de los datos 72
Figura 30 Comprensión de datos. Otra perspectiva 73
Figura 31 Correlación de Karl Pearson 74

Figura 32 Preparación de datos 75
Figura 33 DataSet discretizada y normalizada 76
Figura 34 Set de datos importado de una tabla 76
Figura 35 Muestra datos completos 77
Figura 36 Muestra eliminación del campo Nro 77
Figura 37 DataSet, sin campo Nro. 78
Figura 38 Gráfico de Calor con las dimensiones 78
Figura 39 Grafico indicando las variables que tienen correlación >0.5, reduciendo las dimensiones de 13 a 8 incluida la variable de clase 79
Figura 40 Gráfico de Calor con las dimensiones reducidas de 13 a 8 incluida la variable de clase 79
Figura 41 Data Set definitivo 80
Figura 42 Importación de algoritmos de aprendizaje, métricas y libreras gráficas para la implementación de los modelos 81
Figura 43 DataSet seleccionado 81
Figura 44 Separación del Set de datos 82
Figura 45 Separación del set de datos en entrenamiento y pruebas 83
Figura 46 Separación del Set de Datos en entrenamiento y prueba (filas) 83
Figura 47 Entrenando algoritmo para obtener el modelo 1 84
Figura 48 Entrenando algoritmo para obtener el modelo2 84
Figura 49 Entrenando algoritmo para obtener el modelo3 85
Figura 50 Entrenando algoritmo para obtener el modelo4 85
Figura 51 Muestra los vectores de predicción generados a partir del set de prueba86
Figura 52 Muestra los porcentajes de confiabilidad de cada modelo 87
Figura 53 Modelo ensamblado con Voting Classifier 87
Figura 54 Obteniendo el modelo ensamblado 88
Figura 55 Haciendo una predicción con el modelo ensamblado 89
Figura 56 Predicción con el modelo ensamblado 90
Figura 57 Codificando el modelo ensamblado 91
Figura 58 Prueba de modelo con datos de un estudiante 91
Figura 59 Modelo 1 de Regresión Logística Binaria 92
Figura 60 Matriz de confusión del modelo1 regresión logística binaria 92
Figura 61 Modelo Naive Bayes 93
Figura 62 Matriz de confusión del modelo 2 94

Figura 63 Modelo 3 de Bosques Aleatorios 95
Figura 64 Matriz de confusión del modelo 2 95
Figura 65 Modelo 3 de Clasificador de Soporte Vectorial 96
Figura 66 Matriz de confusión del modelo 4 97
Figura 67 Modelo Ensamblado con Voting 98
Figura 68 Matriz de confusión del modelo Ensamblado 98
Figura 69 fase de despliegue del modelo 100
Figura 70 Ejecución de un programa usando el modelo predictivo Ensamblado, Para verificar que un estudiante no desertará 100
Figura 71 Ejecución de un programa usando el modelo predictivo Ensamblado, para verificar que un estudiante si desertará 101

RESUMEN

El presente trabajo de investigación, se inicia con el estudio de la deserción estudiantil con datos recopilados de estudiantes en 10 periodos académicos del programa Computación e Informática, comprendidos entre el año 2012 y 2021 del Instituto Superior Tecnológico Trujillo. La información recopilada está compuesta de 500 registros de estudiantes. Se planteó el siguiente problema de investigación ¿Cómo obtener un modelo predictivo con mayor porcentaje de confiabilidad que los algoritmos tradicionales Bayesianos, Regresión, Soporte Vectorial y Bosques Aleatorios, para estimar la deserción estudiantil de modo más eficiente en los centros de Educación Superior Tecnológicos Públicos de la región La Libertad a través de máquinas de aprendizaje supervisados?. Se utilizó una metodología de minería de datos conocida como Crisp - DM y algoritmos de aprendizaje supervisado de la librería Scikit Learn de Python. El Tipo investigación es aplicada y diseño descriptivo. Se emplearon cuatro algoritmos de aprendizaje para ser entrenados en un conjunto de datos correspondiente al 80% del set de datos original, reservando 20% restante para la etapa de prueba del modelo final. Se obtuvo cuatro modelos, que luego de ser evaluados, se utilizaron como entradas para generar un nuevo modelo ensamblado mediante un algoritmo de votación usado para este propósito, por tener modelos heterogéneos. Se realizó en ensamble y luego de la evaluación de confiabilidad con el 20% de la data reservada para la prueba, se evidenció una confiabilidad del 93%, con una tasa de errores de 7/100 (7%), entre patrones reconocidos y no reconocidos. Se determinó el modelo ensamblado como propuesta del trabajo de investigación, aun cuando este mismo indicador se presenta en dos modelos básicos analizados; sin embargo, el modelo ensamblado, por su robustez que los caracteriza, es el que se elige como propuesta para garantizar la permanencia del índice de confiabilidad durante todo el ciclo de vida.

Palabras claves

Deserción estudiantil, modelo, predictivo, conjunto de datos, set de datos, aprendizaje automático supervisado, algoritmo de clasificación.

ABSTRACT

The present research work begins with the study of student dropout with data collected from students in 10 academic periods of the Computing and Informatics program, between 2012 and 2021 at the Instituto Superior Tecnológico Trujillo. The information collected is composed of 500 student records. The following research problem was posed: How to obtain a predictive model with a higher percentage of reliability than the traditional Bayesian algorithms, Regression, Support Vector and Random Forests, to estimate student dropout more efficiently in the Public Technological Higher Education centers of the La Libertad region through supervised learning machines? A data mining methodology known as Crisp - DM and supervised learning algorithms from the Scikit Learn Python library were used. The research type is applied and descriptive design. Four learning algorithms were used to be trained on a data set corresponding to 80% of the original data set, reserving the remaining 20% for the final model testing stage. Four models were obtained, which after being evaluated, were used as inputs to generate a new model assembled through a voting algorithm used for this purpose, due to having heterogeneous models. It was carried out as an assembly and after the reliability evaluation with 20% of the data reserved for the test, a reliability of 93% was evident, with an error rate of 7/100 (7%), between recognized and unrecognized patterns. recognized. The assembled model was determined as a proposal for the research work, even though this same indicator is presented in two basic models analyzed; However, the assembled model, due to its robustness, is the one chosen as the proposal to guarantee the permanence of the reliability index throughout the life cycle.

Keywords

Student dropout, model, predictive, data set, data set, supervised machine learning, classification algorithm.

INTRODUCCIÓN

En el mundo actual las máquinas de aprendizaje automático, conocidas como Machine Learning, cuentas con una gama muy amplia de algoritmos de aprendizaje, utilizados en la confección de modelos predictivos y aplicadas en las diferentes áreas de la actividad humana, tales como la empresa, servicios de transporte, en medicina para el diagnóstico de tumores y en muchos campos más incluyendo la educación. Estos algoritmos tienen la capacidad de aprender a partir de una gran cantidad de datos que se le suministre, generalmente históricos poder detectar patrones de comportamiento

El alto grado de deserción estudiantil, se ha convertido en un problema importante para muchas instituciones de educación superior que deben tener en cuenta tanto en nuestro medio como en América Latina y el Caribe. Se considera, que el índice de deserción por año, se encuentra en alrededor del 57%, (Cordera, Arruti, Peralta, Popoca, Sheinbaum, & Victoria, 2007).

En los últimos años, la deserción estudiantil se ha convertido en tema de mucha preocupación para las autoridades de los Institutos Tecnológicos y, a su vez, para otros especialistas. La deserción no solo debe ser materia de estudio del sector educación, sino que debe ser enfocada desde una perspectiva multidisciplinaria que tenga en cuenta el aspecto económico, social y pedagógico.

Según un informe de la UNESCO, en periodos de tiempo normal, se gradúan alrededor del 43% de los estudiantes que ingresan a una carrera profesional (Manos Antoninis, 2020).

La minería de datos, es una disciplina que permiten la incorporación de metodologías de desarrollo de modelos predictivos para abordar problemas de predicción relacionados con cualquier fenómeno que enfrenta nuestra sociedad actual con altos índices de confiabilidad (Timar y Jim, 2015). De esta forma, se puede pronosticar, la probabilidad de deserción de un estudiante, basado en los datos históricos, almacenados en los sistemas de información (Spositto y Etcheverry, 2010), de las instituciones.

El ámbito inicial elegido como referencia para la investigación del presente proyecto, fue el Instituto Superior Tecnológico Publico Trujillo, ubicado en la provincia de Trujillo, departamento de La Libertad, sin embargo, la misma problemática se observa en los institutos superiores tecnológicos públicos de las diferentes regiones, extendiéndose dicho proyecto a todo el ámbito nacional.

Desde hace dos décadas se viene observando que la cantidad de alumnos que ingresan a realizar sus estudios profesionales, al primer periodo lectivo, estos se ven disminuidos al iniciar el segundo periodo lectivo en cantidades considerables; situación preocupante porque, al no poner atención a dicha problemática, esto podría poner en riesgo la continuidadde las instituciones educativas, debido a la deserción.

Este hecho motivó pensar en la posibilidad de poder predecir qué alumnos con cierto gradode confianza podrían ser los desertores o en su defecto determinar los alumnos que tiene riesgo de no continuar estudios el próximo periodo académico y de ser así informar al área de tutoría para una atención personalizada a través de su departamento; llevando esta situación a formularme la presente interrogante: ¿Cómo obtener un modelo predictivo con mayor porcentaje de confiabilidad que los algoritmos tradicionales Bayesianos, Regresión y Árboles de Decisión, para estimar la deserción estudiantil de modo más eficiente, en los centros de Educación Superior Tecnológicos Públicos de la región La Libertad a través de máquinas de aprendizaje?.

Se propone un modelo basado en el uso de técnicas de clasificación en Máquinas de Aprendizaje Supervisadas, tales como Regresión Logística Binaria, Naive Bayes, Bosques Aleatorios y Clasificador de Soporte Vectorial con el fin de predecir la deserción estudiantil,con alto grado de confiabilidad en grupos de alumnos que cursan el primer año académico en los Institutos de educación superior tecnológicos públicos de la región la Libertad.

Se utilizó fichas socioeconómicas, informe de notas, datos históricos del centro educativo en mención para confeccionar el set de datos, se realizó la limpieza de ellos a fin de obtener el conjunto de datos del proyecto. Se dividió el conjunto de datos tanto para entrenamiento como para la prueba. Se realizó el entrenamiento usando cada algoritmo de indicado para el primer conjunto de datos y obtuvo los 4 modelos predictivos, a los cuales se evaluó la confiabilidad de cada uno y determino el mejor, el cual formó parte de la propuesta.

Se utilizó algoritmos o métodos de clasificación supervisados para el entrenamiento que permitan utilizar un gran número de dimensiones independientes, las mismas que representarán las características de los estudiantes, que conforman el set de datos a utilizar.

El objetivo de este estudio es proponer un modelo predictivo ensamblado óptimo basado en Machine Learning Supervised para estimar la deserción estudiantil de los estudiantes

en los centros de Educación Superior Tecnológicos Públicos de la región La Libertad.

El marco metodológico que guiará el desarrollo del presente proyecto es la metodología ágil denominada CRISP -DM, muy utilizada en proyectos de minería de datos por su versatilidad.

CAPÍTULO I PROBLEMA DE INVESTIGACIÓN

1.1. Planteamiento y fundamentación del problema de investigación

1.1.1. El problema en la actualidad a nivel internacional

El problema de la deserción estudiantil, no es un problema nuevo ni local, tiene un ámbito muy grande, sin embargo, esta problemática, tiene mayor impacto en sociedades subdesarrolladas, frenando las oportunidades de crecimiento, aumentando el subempleo y por tanto la mano de obra no calificada, por tal razón es importante abordar el tema a fin de limitar las consecuencias negativas en la sociedad.

Según el Banco Mundial, en el 2018, el 30% de los estudiantes peruanos de nivel superior dejan su carrera por diferentes motivos, del mismo modo el porcentaje en otros países de América Latina llega a 42% de los alumnos que desertan de sus estudios, demostrando así un gran problema social con consecuencias negativas a la sociedad.

La importancia del presente trabajo de investigación radica en la necesidad de realizar estudios orientados a la elaboración de modelos de predicción usando máquinas de aprendizaje supervisadas en el ámbito educativo y en nuestra región de la Libertad; con el propósito de pronosticar eventos futuros de forma más precisa disminuyendo la incertidumbre de la deserción estudiantil y contar con un plan de tutoría efectivo.

1.1.2. El problema en la actualidad a nivel nacional

Según INEI, en el Perú, existen 860 Institutos de Educación Superior Tecnológica, que albergan aproximadamente 1,176,000 estudiantes de los cuales el 16% son estudiantes de zona urbana y el 7,7% son estudiantes de zona rural, cuyas edades oscilan desde los 15 a 29 años, segmento que 2 de cada 10 jóvenes son pobres.

Por otro lado, solo el 4% del PIB es destinado para el sector educación, es utilizado para mejorar la infraestructura de los planteles, currículos nuevos, capacitación docente, desayunos escolares, contribuyendo al mejoramiento de las condiciones generales de calidad del sistema

educativo nacional. Sin embargo, a nivel micro se requieren de políticas educativas y acciones específicas que permitan revertir los niveles de deserción escolar y contribuir reforzando las condiciones para el desarrollo intelectual, psicológico, vocacional y familiar de los estudiantes como factores importantes en la problemática para producir el efecto contrario a la deserción que sería de retención en el sistema educativo nacional.

La deserción estudiantil se ha incrementado significativamente en Perú antes y más aún más después de la pandemia de COVID-19, es por ello, que los Institutos Superiores Tecnológicos Públicos requieren identificar e implementar programas para disminuirla. El presente trabajo de investigación tiene como propósito determinar un modelo que permita predecir el fenómeno de la deserción estudiantil en los institutos superiores tecnológicos de la región con un buen porcentaje de confiabilidad.

1.1.3. El problema actual a nivel local

Según INEI, en el departamento de La Libertad existen 27 Institutos de Educación Superior Tecnológica, uno de ellos es el Instituto Superior Tecnológico Trujillo, donde existe un porcentaje considerable de alumnos que deciden abandonar sus estudios, por diversos motivos, entre ellos económicos, académicos, vocacionales e incluso problemas de adaptación.

Cada año, al iniciar las matrículas del tercer semestre, la cantidad de estudiantes disminuye considerablemente con respecto al año anterior que ingresaron, dicha situación se acentúa más al iniciar el tercer semestre o segundo periodo académico. Lo que origina que los estudiantes pierdan la expectativa de continuar con su carrera profesional técnica, motivo por el cual muchos abandonan sus estudios.

El lugar donde se realizó el recojo de información es el IESTP Trujillo – Trujillo – La Libertad.

Los factores que influyen en la realidad problemática de estudio, son:

nivel socioeconómico, académico de los ingresantes, compromisos familiares, ingreso familiar, carga familiar, desaprobación de materias básicas en el primer semestre entre otras.

La atención oportuna de la presente realidad problemática permitirá actuar demanera eficaz a los actores educativos para evitar la deserción estudiantil en los institutos tecnológicos de la región La Libertad. Situación que disminuiría problemas sociales como el desempleo, por falta de personal calificado.

El presente proyecto de investigación utiliza metodologías y técnicas de Machine Learning para la confección de modelos predictivos basados en los algoritmos de aprendizaje supervisados conocidos como Regresión Logística Binaria, Bosques Aleatorios, Máquinas de Soporte Vectorial y Naive Bayes. Evaluar sus coeficientes de confiabilidad y determinar el mejor de ellos.

1.2. Antecedentes de la investigación

1.2.1. Internacionales

Masabanda & Zapata (2019), basaron su estudio experimental en una encuesta en línea aplicada a 1457 alumnos de la Facultad de Ciencias de la Ingeniería y Aplicadas de las Carreras de Ingenierías: Eléctrica, Sistemas de Información, Electromecánica e Industrial. Usó la metodología KDD (Knowledge Discovery in Databases). Los resultados le permitieron determinar que los factores: conducta, bullying, motivación del docente-alumno, bajo conocimiento de la asignatura, vicio por las redes sociales, estado emocional, conocimiento adquirido en los cursos de nivelación, formación académica, selección en el ingreso a la universidad, problemas familiares, residencia y expectativas respecto a la carrera, son los factores que tienen mayor influencia en la deserción de los estudiantes en la Facultad de Ciencias de la Ingeniería y Aplicadas. Mientras que las técnicas de minería de datos J48,

Random Forest y Sequential Minimal Optimization (SMO), dieron como resultado una tasa de predicción de la deserción del 92%. Se concluyó que el uso de técnicas de minería de datos puede ser consideradas como importantespara realizar estudios de las causales que afectan a los estudiantes en su permanencia estudiantil universitaria. Además, esta herramienta podría ser considerada como una herramienta de apoyo para las autoridades universitarias a fin de que se establezca estrategias y políticas que permitan mitigar las tasas de deserción.

Avila, Mayer y Quesada (2021), aplicaron técnicas de minería de datos basada en Machine Learning, mediante el uso de algoritmos de aprendizaje supervisado que permitan generar modelos de predicción de la deserción estudiantil que de manera temprana determine si un estudiante probablemente desertará de su proceso de formación. Durante el desarrollo de este proyecto se utilizaron herramientas de software Libre tales como WEKA que permitieron obtener algunos resultados a partir de la aplicación de algoritmos de Machine Learning. Se hizo un diagnóstico de la percepción de la deserción en el cuerpo docente mediante la aplicación del instrumento tipo cuestionario lo cual posibilitó obtener datos muy interesantes acerca de cómo se percibe la problemática de deserción en el cuerpo docente, con resultados que permitieron concluir que el cuerpo docente considera que un modelo de predicción de la deserción estudiantil en la UNAD contribuye en gran medida al mejoramiento de la retención y permanencia de los estudiantes.

1.2.2. Nacionales

García (2019), aplicó un modelo de clasificación basado en Machine Learning, para analizar el comportamiento de los estudiantes, teniendo en cuenta factores como cursos

matriculados, cantidad de cursos aprobados, si es independiente o depende de sus padres con respecto al pago de sus pensiones por estudios, si tiene o no sanción disciplinaria por la entidad respectiva, cantidad de cursos desaprobados durante el semestre, cantidad de cursos desaprobados dos veces, cantidad de cursos desaprobados de tres veces a más, número de créditos aprobados, créditos desaprobados, promedio ponderado final, situación regular o irregular de los estudiantes. Logró demostrar la potencia y virtudes que tiene el modelo XGBoost con respecto al pronóstico de estudiantes propensos a abandonar sus estudios en la Universidad Peruana Unión Filial, Juliaca. Demostró que el factor más relevancia para desertar de la universidad, es el número de créditos aprobados que se matriculó en el ciclo académico. Los factores que no afectan mucho en la deserción de un alumno son los estudiantes independientes, también considera a los que tiene una sanción disciplinaria por el área de bienestar universitario. Finalmente, logra demostrar que el algoritmo que se implementó para esta investigación fue XGBoost y que una de muchas ventajas es que trabaja de acuerdo al porcentaje de error que tuvo el árbol anterior y para el siguiente árbol tiene disminuye ese porcentaje de error.

Mamami (2019), empleó la metodología CRISP DM para desarrollar su modelo predictivo y se realizó un primer enfoque de levantamiento de información para analizar las variables más resaltantes para el modelo de clasificación, posteriormente se hizo una exploración de la data para definir la arquitectura del modelo. Conclusión, se desarrolló e implementó un modelo minería de datos aplicando la hiper parametrización para observar el comportamiento de la ANN en la predicción de deserción estudiantil universitaria, se identificó variables de los factores asociados que aportaron en el ciclo del proyecto.

Pérez, Nieto, Moncada, Quinteros, Ortiz y Rojas (2020), usaron el algoritmo de aprendizaje supervisado denominado vectores de soporte vectorial. Para su conjunto de datos usaron información histórica de estudiantes que desertaron en periodos académicos pasados y como algoritmo de entrenamiento máquinas de vectores de soporte para entrenar el modelo. Los autores formularon las siguientes conclusiones: se obtuvo un grado de confiabilidad mayor a 90% superando el grado base que indicaron como base para su hipótesis. El modelo obtenido identifica los factores con mayor influencia de deserción estudiantil debido a que utiliza como base de aprendizaje la información de alumnos que ya desertaron y predice los patrones de comportamiento y por lo tanto el factor con mayor influencia. Se obtuvo los siguientes datos: de 1533 alumnos, se identificaron a 46 desertores (equivalente a 3%) y a 1487 como alumnos no desertores.

Padilla(2019), investigó el problema**,** llegando a la conclusión que el modelode minería de datos propuesto, hace mención a algunas investigaciones que utilizan diversas arquitecturas de algoritmos en el campo educacional como la predicción de cursos a elegir, rendimiento académico, clasificación de estudiantes, entre otros y sostiene que los estudiantes pueden desertar cuando su promedio del semestre es muy bajo, por ello surge el interés de explorar los diversos factores que motiva a una persona a tomar esa decisión yfundamentalmente llevar ese pensamiento a una máquina capaz de realizar laclasificación para generar la misma actividad humana.

Emplea para ello la metodología CRISP DM para guiar el desarrollo del modelo predictivo y realiza un primer enfoque de levantamiento de información y analiza las variables más

resaltantes para el modelo de clasificación, luego hace una exploración de la data para definir la arquitectura del modelo y finalmente evalúa las eficiencia del algoritmo para observar el aprendizaje de la máquina y comparar los resultados con nueva data determinando la influencia del modelo con la estimación de deserción.

1.2.3. Locales

Se realizó una búsqueda en los servidores de las bases de datos de trabajos de investigación y no se encontró ningún antecedente local.

1.3. Formulación del problema de investigación

Teniendo en cuenta lo anterior se plantea el siguiente problema:

¿Cómo obtener un modelo predictivo con mayor porcentaje de confiabilidad que los algoritmos tradicionales Bayesianos, Regresión, Bosques aleatorios y Máquinas de Vectores de Soporte, para estimar la deserción estudiantil de modo más eficiente, en los centros de Educación Superior Tecnológicos Públicos de la región La Libertad a través de máquinas de aprendizaje?

1.4. Delimitación del estudio

El ámbito elegido como referencia para la investigación del presente proyecto fue el Instituto Superior Tecnológico Público Trujillo del distrito de Trujillo, provincia de Trujillo, departamento de La Libertad, sin embargo, la misma problemática se observa en los Institutos Superiores Tecnológicos Públicos de la región La Libertad, pudiendo extenderse dicho proyecto a otras regiones del ámbito Nacional.

En la presente investigación, el modelo predictivo es el esquema que se ajusta mejor a la problemática de la deserción estudiantil y podrá identificar los patrones de posibles estudiantes que desertarían en este

escenario. Se tiene como alcance sistematizar el fenómeno de la deserción a través del modelo predictivo propuesto. Del mismo modo la presente investigación, dirige sus esfuerzos en la importancia de tener una aplicación informática que de soporte a determinar la deserción de estudiantes dentro del centro educativo. El proyecto usa la base de datos de estudiantes que han desertado antes de la ejecución de este modelo tecnológico. Aclarando lo siguiente: que la presente investigación no contempla la fase de despliegue o implementación del modelo predictivo y solo se centrará en el diseño del modelo.

Limitaciones

El presente trabajo de investigación recopiló información de las fichas socioeconómicas de los 10 últimos periodos académicos, información con la que se cuenta actualmente.

1.5. Justificación e importancia de la investigación

Entendiendo que la investigación científica es el instrumento intelectual que durante mucho tiempo ha permitido la generación de conocimiento, la presente investigación crea un modelo predictivo que, a partir de datos característicos de un estudiante, el modelo infiere y crea conocimiento, como es la predicción ante la incertidumbre, de si un estudiante abandonara o no sus estudios en el próximo semestre académico dado un conjunto de características.

Por tanto, en este trabajo de investigación se considera que las propuestas de modelos predictivos basados en algoritmos de aprendizaje supervisados, son la mejor alternativa tecnológica actual que se ajusta a la problemática de la deserción estudiantil. A través de estos modelos, las instituciones podrán identificar a los estudiantes más probables de abandonar sus estudios con un buen índice confianza.

En tal sentido, la aplicación del modelo predictivo aportará una nueva forma de estudiar la información de la deserción y así disminuir sus

altos índices actuales.

El modelo predictivo propuesto tiene la capacidad de identificar patrones de estudiantes que desertarían y convertir estos datos en información y conocimiento valioso para la institución.

Con referencia a la importancia de la investigación, esta aporta a la sostenibilidad estudiantil a lo largo de los periodos académicos en las instituciones educativas superiores tecnológicas del país, permitiendo a estas cumplir las condiciones básicas de calidad estipuladas por SINEASE y mantenerse dentro de las instituciones licenciadas y acreditadas.

Esta investigación basó su estudio en la implementación de cuatro modelos y determina el mejor modelo de ellos; es decir, aquel que tenga mayor grado de confiabilidad, el mismo que permitirá en los estudiantes ingresantes al primer año de estudios de los Institutos Superiores Tecnológicos, identificar a aquellos con posible riesgo de abandono de la institución y por tanto brindar una atención personalizada a través del área de tutoría y disminuir en los posible la deserción y garantizar la sostenibilidad de los estudiantes y cumplir así con uno de los factores de calidad institucional.

Desde el punto de vista científico, a través del presente proyecto se, refuerza el uso de aplicaciones predictivas en fenómenos relacionados con la deserción estudiantil a través de métodos de selección, usando algoritmos de aprendizaje supervisado.

Estos métodos han sido estudiados hace poco más de 200 años, tales como los métodos de regresión lineal propuestos por Legendre 1805 y mejorados por Gauss en 1809, para determinar la curvatura elíptica de los objetos que giran alrededor del sol; se llamaba método de mínimos cuadrados, posteriormente en 1900 se utiliza el término de regresión lineal, del mismo modo los métodos bayesianos, árboles de decisión, J48 entre otros.

La finalidad del modelo predictivo, aplicado a la deserción estudiantil es predecir la cantidad y quienes desertarían sus estudios superiores,

sin un conocimiento de pronósticos sobre la problemática. El modelo será alimentado con una gran cantidad de datos (Big Data) históricos de 10 años atrás.

Desde el punto de vista académico, la presente investigación, mejorará el trabajo que se realiza en el área de tutoría, acerca de los estudiantes con riesgo de abandonar sus estudios. A su vez servirá de base para futuros proyectos de desarrollo e implementación de Machine Learning, para planes de contingencia institucionales.

Desde el punto de vista Social, la presente investigación tiene un impacto muy importante en la sociedad, debido que, al haber menos deserción estudiantil en los centros de educación superior tecnológica; en el corto plazo, la sociedad contaría con mayor mano de obra calificada que la actual para enfrentar los nuevos retos de las organizaciones y por tanto una disminución en la tasa de desempleo de mano de obra calificada.

1.6. Objetivos de la investigación

1.6.1. General

Determinar un modelo predictivo ensamblado, basado en Machine Learning Supervised, para estimar la deserción estudiantil en los centros de Educación Superior Tecnológicos Públicos de la región La Libertad.

1.6.2. Específicos

1. Obtener el juego de datos inicial de la situación problemática.
2. Preparar el juego de datos para el entrenamiento y prueba de los algoritmos de clasificación.
3. Crear los modelos predictivos de entrada.
4. Ensamblar el modelo propuesto.
5. Determinar la prueba de confiabilidad del modelo propuesto.

CAPÍTULO II MARCO TEÓRICO

2.1. Fundamentos teóricos de la investigación

Un marco teórico es el apartado en el que se exponen los antecedentes, las principales teorías y conceptos que sustentan un proyecto o investigación. Puede incluir también los argumentos e ideas que se han desarrollado en relación a un tema (Ramos, 2018).

Modelo

Un modelo es una representación concreta o abstracta, que al suministrar datos de entrada produce información de diferentes niveles de complejidad basados en parámetros establecidos (Gironés, Casas, Minguillón y Caihuelas, 2017)
"Podemos entender el modelo como la habilidad de aplicar una técnica a un juego de datos con el fin de predecir una variable objetivo o encontrar un patrón desconocido" (Girones et al.,2017)

Un modelo es la representación de un componente o proceso respecto a una parte de la realidad. Se clasifican en modelos estáticos y dinámicos, en los primeros la entrada y salida corresponde a un mismo instante en el tiempo, mientras que los segundos, la salida corresponde a un tiempo distinto al dato de entrada (Maguire, Batty et al., 2005).

Modelo Predictivo

El modelado predictivo, es una técnica matemática que utiliza estadísticas para la predicción. Los modelos predictivos buscan trabajar sobre la información proporcionada para llegar a una conclusión final después de que se haya desencadenado un evento (Girones et al.,2017)

Preparación de datos (Girones et al.,2017)

Las actividades de preparación de datos en investigaciones basadas en minería de datos o de Machine Learning se dirigen a la conformación de un set de datos para ser usados posteriormente por algún algoritmo ya sea de clasificación, segmentación o regresión.

Estas actividades consisten en:

1.- Limpieza de datos: este proceso implica detección, eliminación o corrección de conjuntos de datos corruptas o inapropiadas en los Set de datos. Se gestionan valores ausentes, erróneos o inconsistentes.

Durante la integración de los datos, esto pude ser una fuente de incoherencias en los datos que deben ser detectados y subsanados.

2.- Normalización de datos: consiste en modificar los datos para un logro de valores que simplifique su comparación entre ellos. Este proceso es útil para varios métodos de minería de datos que muchas veces son sesgados por atributos con valores muy altos, ocasionando que el modelo se distorsione.

Entre algunos métodos de normalización se tiene: normalización por el máximo, normalización por la diferencia y normalización basada en la desviación estándar.

3.- Discretización de datos: mediante este proceso los valores de una variable continua se incluyen en categorías o intervalos o grupos con el fin de limitar los estados posibles.

La discretización permite disminuir el costo computacional y aumenta el proceso inductivo.

4.- Reducción de la dimensionalidad: en la implementación de algunos modelos la gran cantidad de datos con que cuenta el set de datos, el tiempo requerido para su procesamiento muchas veces no es el adecuado disminuyendo el rendimiento del modelo, en estos casos se aplica una serie de operaciones con el fin de facilitar dos objetivos: la reducción del número de atributos a considerar y la reducción de número de casos.

Cualquier operación que se utilice debe asegurar que se mantenga la calidad del modelo resultante.

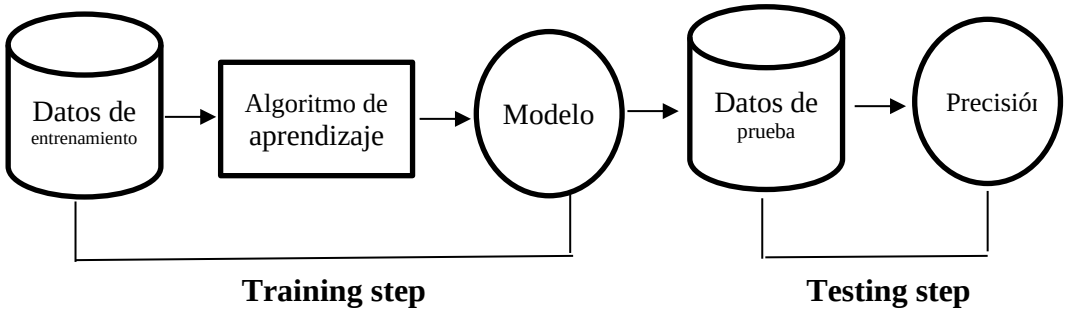
Entrenamiento y test (Girones et al.,2017)

El propósito del entrenamiento y prueba es lograr que los modelos desarrollados a partir del set de datos funcionen correctamente para nuevos datos que hay que procesar, para tal caso el set de datos original debe ser dividido en conjunto de entrenamiento y conjunto de pruebas.

Para validar un modelo se debe asegurar que este funcionará correctamente para los datos de prueba, de tal forma que capture la esencia del problema a resolver y generalice correctamente. Lo que se trata es de evitar que sea dependiente de los datos utilizados en su entrenamiento y evitar el problema conocido como sobre entrenamiento.

Figura 1

Ciclo de vida de un modelo predictivo



Fuente: Girones et al.,2017

Evaluación de modelos (Girones et al.,2017)

La calidad de los modelos predictivos, se realiza comparando las predicciones generadas por el modelo con las etiquetas de clase verdaderas de las instancias del conjunto de datos de prueba. Una mejor vista de este análisis de resultados, se hace a través de una métrica conocida como matriz de confusión.

Matriz de confusión: es una métrica compuesta por una tabla que visualiza de modo gráfico los errores y aciertos logrados por el modelo. Visualiza el nivel de acierto de un modelo de predicción y se conoce también como matriz de errores. (Girones et al.,2017)

Los parámetros que indica son:

- Verdadero positivo
- Verdadero negativo
- Falso negativo
- Falso positivo

Esta métrica es muy utilizada en árboles de decisión, Naive Bayes y máquinas de soporte vectorial (Girones et al.,2017)

Modelos predictivos basados en aprendizaje automático

Estos modelos actualmente dan soporte al conocimiento y experiencia profesional, reduciendo la subjetividad en muchos casos, tales como los sistemas expertos basado para el diagnóstico médico de enfermedades. Se suelen integrar métodos predictivos para reducir sesgos y subjetividad, a la vez que puede proveer potencialmente nuevo conocimiento médico (Pineda, 2022).

¿Qué es la Inteligencia Artificial?

Es la disciplina informática que consiste en lograr que las máquinas imiten el funcionamiento de los procesos naturales humanos. A nivel de negocio la IA nos permite aprender de nuestro volumen de datos.

"La Inteligencia Artificial es una tecnología que permite a las máquinas percibir analizar y aprender del entorno". Con esta información predicen y toman sus propias decisiones para alcanzar sus metas específicas. En esencia la IA es un conjunto de técnicas que imitan la inteligencia del ser humano.

En un artículo publicado, en el 2004, Jhon McArthy, siendo el pionero de la IA, define IA como la ciencia y la ingeniería de crear máquinas inteligentes, es decir programas informáticos inteligentes.

Luego de los aportes de Jhon McArthy y Alan Turing en este campo de la IA, IBM la define como una disciplina que combina la informática y los juegos de datos, permitiendo la resolución de problemas y abarcando sub categorías, como Machine Learning y DEEP Learning. (Gerón, 2019)

Inteligencia Artificial en la ciencia de datos (Joyanes, 2023)

Desde la llegada desde Big Data la IA está presente en numerosos sectores donde antes era prácticamente impredecible que sucediera, lo real es que está impactando en el desarrollo de esta nueva disciplina emergente denominada Ciencia de datos en organizaciones y empresas.

Desde que Jhon McArthy acuña el término de IA en la década del 50, ésta a sufrido una gran evolución que se manifiesta en la década actual con el crecimiento de grandes volúmenes de datos y consolidación de la Ciencia de Datos como campo multidisciplinar. Sin embargo, el aprendizaje automático (Machine Learning) sigue siendo una de las técnicas más utilizadas de la IA en la actualidad. Sin descartar las múltiples aplicaciones que se vienen desarrollando con aprendizaje profundo (DEEP LEARNING) y las redes neuronales, así como el procesamiento del lenguaje natural. Por lo tanto, la IA a logrado integrarse completamente en la ciencia de los datos.

Machine Learning

Disciplina científica y arte de programar computadoras con el fin de aprender dado un conjunto de datos. Campo de estudio que suministra a los ordenadores la capacidad de aprender de manera explícita (Gerón, 2019).

El aprendizaje automático es una sub disciplina de la informática que usa métodos matemáticos y estadísticos para desarrollar soluciones con capacidad de aprendizaje a partir de un juego de datos. En esta investigación usaremos el término aprendizaje para referirnos a la detección y extracción de patrones observado en el juego de datos para el aprendizaje. Los datos por sí no significan nada pero el aprendizaje sobre ellos permite extraer conocimiento. Por tanto, un entendimiento más preciso de la realidad y tomar mejores decisiones (Caballero, Martìn & Riesco, 2019)

El aprendizaje automático, no es nuevo, data de varias décadas atrás, recientemente a llegado a recibir un interés especial al aparecer una disciplina emergente conocida como ciencia de datos.

Las razones de este crecimiento es el incremento de la capacidad de almacenamiento y computación en la nube, y el crecimiento de datos disponibles a partir de sensores, dispositivos móviles y aumento de redes sociales (Caballero, Martín & Riesco, 2019).

Aplicaciones de Machine Learnig (Gerón, 2019)

1. **Clasificadores de imágenes de productos de una cadena de producción,** esta clasificación se realiza utilizando las redes neuronales convolucionales

2. **Escaneos cerebrales para la detección de tumores.** Se usa segmentación semántica y se clasifica cada pixel de la imagen para determinar su ubicación y tamaño exacto del tumor
3. **Clasificador de artículos de noticia automático.** Usa procesamiento de lenguaje natural y clasificación de texto usando redes neuronales recurrentes.
4. **Creación de asistentes personales o chatbot.** Usa procesamiento de lengua natural y para la comprensión del lenguaje módulos de preguntas respuestas.
5. **Segmentar clientes en función de sus compras para diseñar estrategias de marketing por segmento.** Utiliza algoritmos de agrupamientos.
6. **Crear un bot inteligente para un juego.** Utiliza aprendizaje por refuerzo entrenando agentes para la elección de acciones que maximizan la recompensa con el tiempo, dentro de un contexto dado.
7. **Asistentes digitales*:*** Son aplicaciones digitales trabajan con PLN, son aplicaciones de Machine Learning, las cuales permiten a las computadoras procesar texto, voz y comprender el lenguaje humano tal como las personas. Entre ellos tenemos a Apple asistentes, Siri, Amazon Alexa, Google Assistant y otros.
8. **Detección del fraude:** los algoritmos de aprendizaje usados en regresión y clasificación en Machine Learning han sustituido a los sistemas de detección de fraudes basados en reglas, ya que tienen un alto número de afirmaciones falsas al marcar el uso de tarjetas de crédito robadas y casi nunca detectan el uso delictivo de datos financieros robados o vulnerados.
9. **Ciberseguridad**: En el campo de la ciberseguridad, el Machine Learning extrae información de informes de incidentes, alertas, publicaciones en blogs, para identificar amenazas potenciales, asesorar a analistas de seguridad.
10. **Vehículo autónomo**: Estas aplicaciones están equipando actualmente muchos de los vehículos modernos, el cual permite identificar continuamente objetos en el entorno, guía el vehículo alrededor de los objetos, así como el destino del conductor. Muchas formas de Machine Learning, están jugando un papel en la fabricación de vehículos autónomos.

Tipos de Machine Learnig (Gerón, 2019)

Criterios de clasificación:

- ✓ Si el entrenamiento es o no bajo supervisión humana. Entre ellos tenemos: Aprendizaje supervisado, Aprendizaje semi supervisado y aprendizaje por refuerzo.
- ✓ Si aprenden o no de forma gradual durante la ejecución: aprendizaje en línea frente a aprendizajes por lotes.
- ✓ Si funcionan comparando datos nuevos con datos conocidos o si detectan patrones en datos de entrenamiento creando modelos predictivos como hacen los científicos

Métodos de Machine Learning

Los métodos de Machine Learning se dividen en tres categorías principales.

***1.*- Aprendizaje supervisado**

El aprendizaje supervisado se encarga de definir patrones en un juego de datos denominado de entrenamiento, con el propósito encontrar atributos que servirán como plantillas de datos a partir de los cuales poder realizar predicciones en un nuevo juego de datos. Se le denomina supervisado porque el modelo infiere información a partir de un algoritmo y un conjunto de datos etiquetado previamente y luego transferir sus características a una predicción (Moor, 2006).

***2.*- Aprendizaje no supervisado**

En el aprendizaje no supervisado, el algoritmo, detecta patrones en un juego de datos sin la necesidad de haber incluido información etiquetada previamente. Es muy usado en tareas de agrupación, asociación y detección de anomalías (Hastie et al., 2009; James et al., 2013).

***3.*- Aprendizaje semi supervisado**

El aprendizaje semi supervisado brinda un término medio entre el aprendizaje supervisado y no supervisado. Durante el entrenamiento, emplea un conjunto de

datos etiquetados más pequeños para orientar la clasificación y la extracción de características de un conjunto de datos más grande y sin etiquetar. El aprendizaje semi supervisado puede solucionar el problema de no tener suficientes datos etiquetados (o no poder permitirse etiquetar suficientes datos) para entrenar un algoritmo de aprendizaje supervisado (Hastie et al., 2009; James et al., 2013).

4.- El aprendizaje por refuerzo

En cuanto al aprendizaje por refuerzo, se trata de una técnica en que la máquina, o agente, recibe una valoración basada en el desempeño de la tarea que ha realizado. En términos generales, se establece un proceso de decisión en el que el agente debe ser capaz de seleccionar una acción determinada que pueda maximizar la obtención de un logro, a partir de un determinado comportamiento (Sutton & Barto, 1998; Russel & Norvig, 2002).

Se espera que el sistema indague su entorno y observando los resultados de algunas acciones y obtener datos a partir de los cuales aprender. Este proceso se realiza sin conocimiento previo de una opción correcta. Se realiza mediante, mediante el ensayo y error (López-Boada et al., 2005).

Combinación de clasificadores o ensamble de modelos (Girones, 2017)

La combinación de clasificadores o también conocido como ensamble de modelos, consiste en crear modelos complejos a partir del ensamble de modelos obtenidos de algoritmos básicos. De tal modo que la decisión tomada por el modelo ensamblado sea una combinación de cientos de decisiones parciales. Los clasificadores usados como base deben ser los más variados posibles con el propósito de que los errores cometidos por el clasificador base sean mínimos con respecto al resto.

Los métodos de ensamble son procesos mediante el cual se construyen varios modelos para resolver un problema particular.

El objetivo de los modelos ensamblados no es lograr una mejor precisión, sino una precisión ROBUSTA.

Existen dos formas en la construcción de ensamble de modelos. La primera forma

combina algoritmos de clasificación trabajando simultáneamente y todos los clasificadores utilizan el mismo algoritmo de entrenamiento, siendo necesario dividir el set de datos original en varios sub DatSet y tomar una decisión conjunta a partir de la decisión parcial de cada uno de ellos. La segunda forma consiste en combinar modelos base diferentes, generados por diversos algoritmos de aprendizaje heterogéneos de forma secuencial, de tal modo que cada clasificador utilice los resultados de un clasificador anterior capturando una característica clave de los datos.

Entre las técnicas de ensamblaje más utilizadas tenemos: clasificador Bagging y clasificador Voting. **(Girones, 2017)**

Voting, es una técnica de ensamble de clasificadores heterogénea, es decir emplea diferentes algoritmos de clasificación para un mismo Set de Datos. Usa como entradas una lista de registros con el nombre, la clase del clasificador y un método: Hard Voting o Soft Voting. Usa técnicas simples para una solución compleja óptima. Esta técnica es la más fácil de entender por ser democrática. La forma de resolver la votación, está en función de si el problema es de clasificación o de regresión.

En el esquema usado como método *hard,* la clase ganadora será la que tanga el máximo número de votos. Se conoce también como votación democrática.

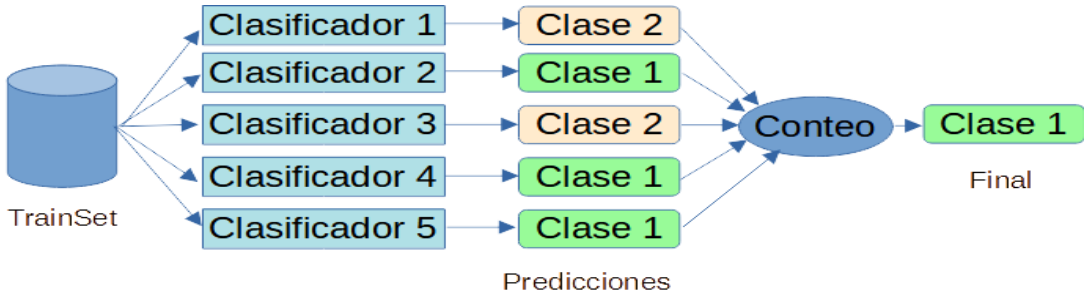
Cuando el esquema hace uso del método soft v*oting, el algoritmo, toma en cuenta* las probabilidades de cada clase. De cada clasificador se obtiene la probabilidad de la clase1, clase2, clase3, etc; luego se promedian los valores de las probabilidades de cada clase y elige la clase que tenga el máximo valor promedio.

Si el problema a solucionar, fuera de regresión, el valor final sería el promedio

de todos los valores definidos por los modelos.

Figura 2

Sistema típico de voting



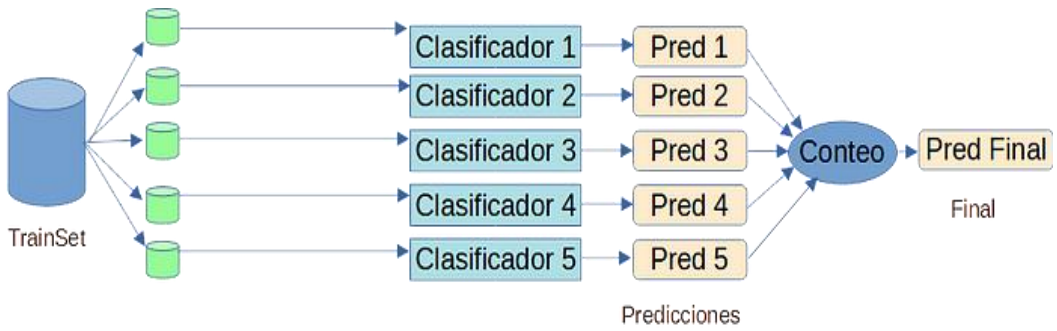
Fuente**:** Girones, 2017

Bagging, es una técnica de ensamble de clasificadores homogéneos, es decir utiliza un tipo de clasificador para diferentes conjuntos de datos. Los conjuntos de datos se obtienen aleatoriamente del conjunto de datos original. El algoritmo de clasificación se entrena con cada uno de los conjuntos de datos seleccionados de forma separada y se promedia los resultados para la solución final.

En Bagging se usa el mismo algoritmo de aprendizaje para generar todos los modelos, lo importante aquí es que los subconjuntos de entrenamiento, son seleccionados de modo aleatorio.

Figura3

Esquema del proceso Bagging



Fuente: Girones, 2017

Para el proceso de obtención de los subconjuntos de datos en baggin, existen tres técnicas

obtener los conjuntos de datos de entrenamiento. Entre ellos: Bootstrapping, Pasting, Random Subspace.

Bootstrapping

La filas del data set son seleccionadas aleatoriamente y con reemplazo, lo que significa que cada fila puede ser seleccionada más de una vez.

Pasting

Las filas del data set son seleccionadas aleatoriamente y sin reemplazo, lo que significa que cada fila puede ser seleccionada solamente una vez.

Random Subspace

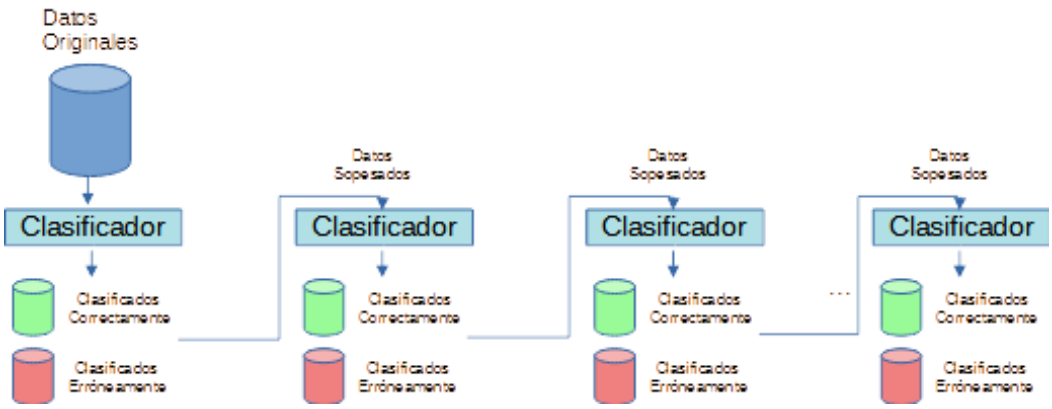
En esta técnica los subconjuntos se seleccionan haciendo proyecciones de las columnas para la obtención de cada sub conjunto de datos; es decir cada conjunto se eligiendo aleatoriamente un determinado número de columnas.

Boosting

El algoritmo boosting consiste en usar varios modelos simples ya partir de ellos obtener un modelo fuerte que tenga menos error. Los modelos se ejecutan de modo secuencial, donde cada modelo de la secuencia toma en cuenta el error de su antecesor. Se asignan pesos a los modelos como se van ejecutando. Los pesos son determinados en función de su desempeño.

Figura4

Esquema del proceso Boosting



Fuente: Girones, 2017

Existen diferentes algoritmos de Boosting. Los más populares son: AdaBoost, Gradient boosting y XGBoost.

AdaBoost

Significa adaptative boosting, trabaja de manera iterativa e identifica las clases que no se clasifican correctamente, luego ajusta los pesos para reducir el error en la etapa de entrenamiento y así se va repitiendo hasta obtener un predictor fuerte.

Gradient boosting

Agrega predictores de modo secuencial. Cada uno corrige los errores de sus predecesores. Respecto a AdaBoost, gradient boosting no cambia los pesos de los puntos, sino que entrena el algoritmo en el error residual del predictor anterior.

XGBoost

Usa entrenamiento paralelo y es una variante del método Gradient boosting. Fue creado para trabajar a mayor velocidad con set de datos muy grandes, para proyectos de big data, usando multiples cores de CPUs.

Deserción Estudiantil

Deserción

Himmel (2002), en base a diversos estudios a nivel internacional, publica un artículo sobre la deserción y la define como el abandono de los estudios sin obtener el título o grado, además, considera un tiempo prudente para descartar la posibilidad de que el estudiante retome sus estudios. Asimismo, destaca dos tipos de deserción: voluntaria y no voluntaria.

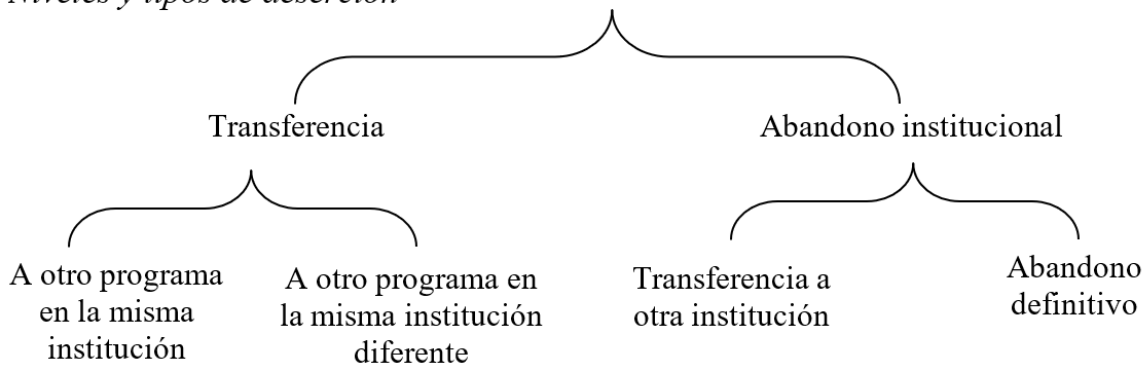
La deserción voluntaria ocurre cuando el estudiante abandona o renuncia a los estudios sin informar a la Institución Educativa.

La deserción no voluntaria se realiza cuando la institución educativa en base a su reglamento, por causas académicas o disciplinarias toma la decisión de separar al estudiante del servicio educativo.

A continuación, se grafica los niveles y tipos de deserción:

Figura 5

Niveles y tipos de deserción



Nota: Fuente: (Himmel, 2002).

Cabe resaltar que el artículo de Himmel es utilizado como inicio de varios estudios nacionales sobre la deserción (Barrios, 2013; de Magalhaes L-Calvet, 2013; Díaz, 2008; Ministerio de Educación, 2012).

Modelos Teóricos de la Deserción

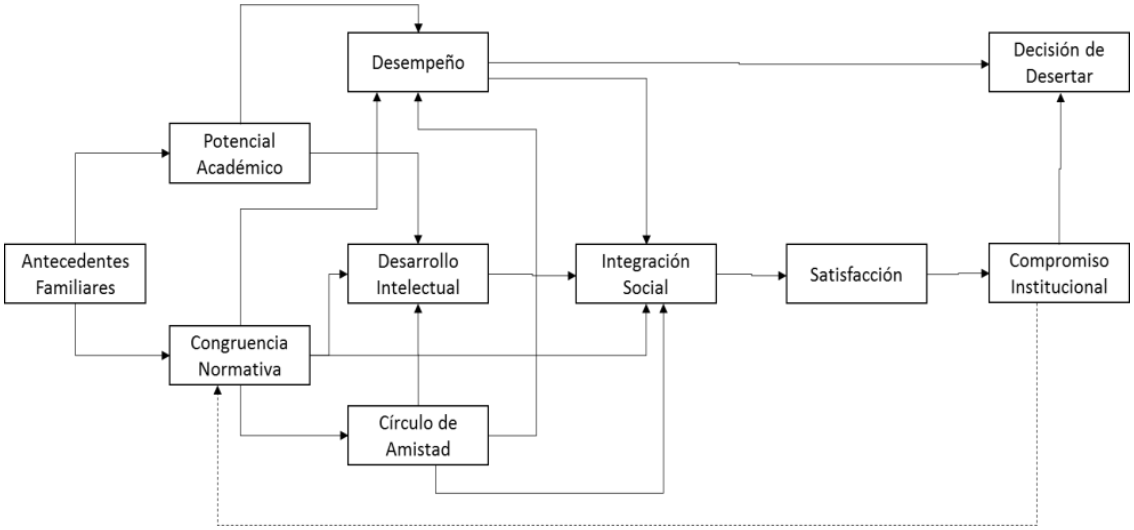
Con la finalidad de explicar la deserción de manera cíclica, existen tres modelos teóricos que son utilizados por los especialistas en el campo de la deserción:

1970: Spady y su modelo basado en la teoría del suicidio

El primer trabajo sobre deserción es implementado por Spady, para ello emplea los principios del suicidio de Durkheim, del que se desprende la decisión de suicidarse. Por lo tanto, la deserción debe explicarse por factores individuales (Durkheim, 1951). Siguiendo este orden, Spady afirma que la deserción sería un resultado de la no integración del individuo con su entorno educativo y deja ver que el ambiente familiar y sus características influyen fuertemente en el estudiante.

Figura 6

Modelo predictivo planteado por Spady



Nota: Fuente: (Spady, 1970a)

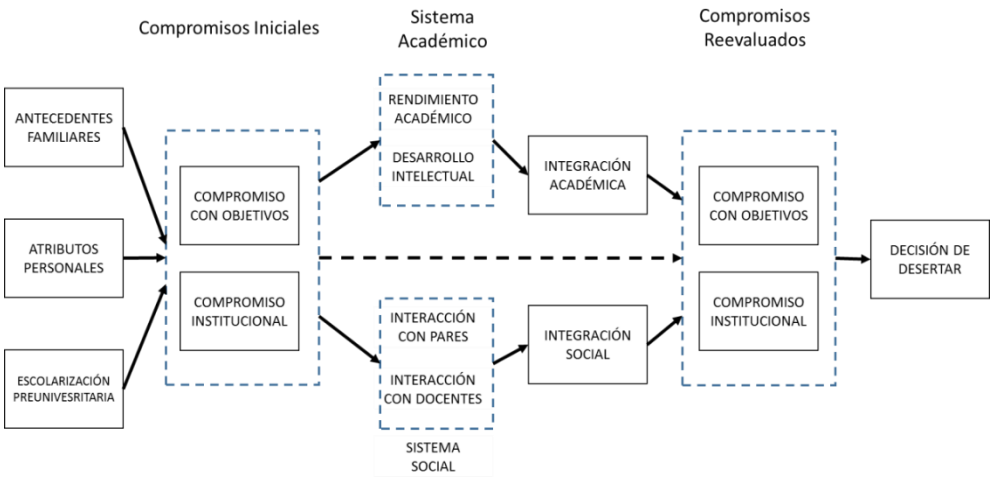
En la **Figura 6**, se plasman todos los elementos que influyen en la decisión final para desertar en cualquier de sus dos formas (voluntaria o involuntaria).

1975: Tinto y su modelo basado en la teoría del intercambio

Tinto (1982), plantea un modelo basado en la incorporación de la teoría del intercambio desarrollado por Nye y sostiene que los estudiantes siempre buscan tener beneficios en la interacción con sus amigos y mejorar así sus estados emocionales (Nye, 1976). Bajo este enfoque, Tinto afirma que los estudiantes van a permanecer en el sistema educativo cuando encuentran mayores beneficios y que aún superen el esfuerzo y dedicación. Pero, el estudiante puede desertar en caso exista otra actividad que le brinde mejores beneficios. En la **Figura 7**, se observa que los estudiantes planifican sus compromisos y metas con la institución educativa y que estos influyen en sus referencias familiares, atributos personales y su experiencia académica. Finalmente, el estudiante realiza una evaluación y reevaluación de sus compromisos y metas personales que podrían desencadenar una deserción si el estudiante siente que los costos son mayores que los beneficios.

Figura 7

Modelo planteado por Tinto



Nota: Fuente:(Cullen & Tinto, 1975).

1985: Bean y su modelo basado en la productividad del ambiente laboral

En 1980, Bean formuló un conjunto de variables que describen la deserción estudiantil, como son: estado económico, desempeño académico previo y residencia actual; las mismas que impactarían en la toma de decisión de desertar (Bean, 1980).

Tabla 1

Lista de variables planteadas por Bean

Variable	Definición
Calidad Institucional	Grado en que la institución educativa es vista como un servicio de buena educación.
Integración	Grado en la frecuencia de sus interacciones con sus pares, en sí si llega a tener amigos cercanos.
Promedio de Notas Universitario	Grado en que un estudiante demuestra su capacidad para desempeñarse en una IES.
Compromiso de Metas	Grado en que obtener un grado universitario es percibido como importante.
Comunicación (Requerimientos/Reglas)	Grado en que la información sobre ser un estudiante es bien vista o no.
Justicia Distributiva	Grado en que un estudiante es atendido por la institución. Por ejemplo: recibe premios y castigos proporcionalmente a su esfuerzo, dedicación o rol realizado como estudiante.
Centralización	Grado en que un estudiante participa en los procesos de toma de decisión. Por ejemplo: centros de estudiantes, consejeros, otros.
Advisor	Grado en que un estudiante cree que su *advisor* es útil.
Relación con funcionarios	Nivel de contactos informales con los miembros de la facultad.
Trabajo en el Campus	Necesidad de tener un trabajo en el campus universitario para permanecer en la escuela
Mejor (área)	El área de uno de los campos de estudio
Mejor(certeza)	Grado en que un estudiante es poco indeciso en que se está especializando
Alojamiento	Cuando una persona vive *On Campus*
Organización del Campus	El número de miembros en la organización del campus
Oportunismo (Transferencia/Trabajo/Hogar)	Grado en que un rol alternativo (como estudiante, empleado o dependiente en casa de los padres) existe en el ambiente externo (otra universidad, en una empresa o volver a casa de los padres).

Variables de Intervención	
Satisfacción	Grado en que siendo un estudiante es visto positivamente
Compromiso Institucional	Grado de lealtad hacia la pertenencia del estudiante en la organización

Nota: Fuente: (Bean, 1980)

Nota: Fuente: (Bean, 1980)

Para el autor, los estudiantes que tienen premios de excelencia académica, tienen mejores desempeños en el instituto; situación que origina el aumento progresivo del grado de satisfacción y compromiso institucional. Resultado de ello, el estudiante no deserta.

En menor grado, se considera relevante las variables de lugar de donde proviene el estudiante, mientras más lejos de su ciudad, menor conexión con esta tenía.

Luego de revisar los modelos de (Spady, 1970),(Tinto, 1975), (Bean, 1985), en la Figura 7, podemos identificar las variables o categorías que tienen alta incidencia tanto para la retención como para el abandono estudiantil universitario:

Figura 8

Categorías o variables que inciden en la persistencia o abandono estudiantil

1970
Spady
Antecedentes familiares
Potencial académico
Congruencia normativa
Desempeño académico
Desarrollo intelectual
Apoyo de pares
Integración social
Satisfacción
Compromiso institucional

1975
Tinto
Antecedentes familiares
Destrezas y habilidades
Rendimiento académico previo
Expectativas de éxito
Compromiso institucional
Desempeño académico
Interacciones con profesores
Actividades extracurriculares
Integración social

1985
Bean
Desempeño académico
Integración académica
Expectativas de éxito
Interacción con pares
Interacciones con profesores
Financiamiento
Integración social
Compromiso institucional

Fuente: Nandeshwar, Menzies, & Nelson, 2011

Según, Nandeshwar, Menzies, & Nelson, 2011, entre ellos utilizaron 15 técnicas de predicción, que va desde regresión múltiple hasta redes neuronales, siendo la técnica más utilizada regresión logística con 57%, siguiéndole con 21% cada uno, la regresión múltiple, análisis discriminante y redes neuronales, las otras técnicas eran no significativas para esta muestra, como se puede ver en la Tabla 2.

Tabla 2 *Técnicas de predicción utilizadas (1971-2008)*

N°	Ténicas utilizadas	Número	%
1	Logistic regression	8	57
2	Multiple regression	3	21
3	Discriminate analyzes	3	21
4	Neural network	3	21
5	C4.5	2	14

Nota. Adaptado de (Nandeshwar et al., 2011)

La retención y abandono en Latinoamérica y el Perú

Ante la problemática aún sin resolver, y a pesar de la gran diversidad de modelos estudiados sobre la retención y la deserción, además teniendo en cuenta los grandes cambios tecnológicos en los últimos tiempos. Observamos que en Latinoamérica es muy reducida la información encontrada. A continuación, mencionamos algunos ejemplos:
En el sistema educativo de Colombia se trabajan 8 ejes para la gestión de permanencia y graduación estudiantil: trabajo colaborativo, compromiso del núcleo familiar, posicionamiento y formalización, cultura de la información, mejoramiento de la calidad. En Chile por su parte, (Paredes Esparza, Aguirre Larrain, & Quense Abarzúa, 2017) en el sistema educativo proponen 4 dimensiones: diagnóstico, desarrollo de habilidades de aprendizaje, apoyos académicos extracurriculares y, acompañamiento y apoyo integral, a los cuales de manera transversal se le tiene que realizar seguimiento, evaluación y replicabilidad, reto que están siguiendo para su implementación total.

Deserción estudiantil en el IESTP Trujillo

En nuestro caso de estudio, en el IESTP Trujillo, como en muchas instituciones de nuestro país, no se aborda el tema de la deserción estudiantil, sin embargo, este es una preocupación a medida que se inician los periodos académicos porque es muy notorio que la cantidad de graduados es muy baja con respecto al ingreso de estudiantes. Situación que solo queda allí en preocupación, no se toman medidas para prevenir dicho fenómeno, aun cuando se cuenta con área de tutoría, pero no tiene acciones que puedan tomar

dicha problemática de modo coherente.

Los informes estadísticos sobre el desempeño académico queda reducido sólo en cifras, además se tiene en cuenta el Reglamento de la Actividad, que entre las actividades no lectivas se encuentra la Tutoría y Consejería que es obligatoria para todos los estudiantes, donde el docente tutor orienta al estudiante en sus actividades académicas y de haber problemas los deriva dependiendo del caso al especialista correspondiente, hecho del cual existe poca información histórica de las incidencias, de haber existido. Luego, el docente emite su informe semestral el cual es derivado a la Jefatura de Departamento Académico del programa de estudios quien, evalúa esta labor docente y eleva un informe periódico al Jefatura Académica y este a su vez lo remite al director general del Instituto. Esos informes, terminan finalmente en el archivo como información histórica para que alguien en algún momento lo requiera. Son pocas los programas de estudios que toman en cuenta esta problemática por esta razón el porcentaje de graduados oscila entre 20 y 25 % de titulados con respecto al número de ingresantes.

1. **Regresión logística.**

   El modelo de regresión logística se entrena y prueba en muestras conocidas. Establecer una relación no lineal entre una variable dependiente y varias variables independientes correspondientes. Relación de respuesta dinámica lineal, y luego ocurre un evento en una muestra desconocida. Los valores de probabilidad se utilizan para predecir o evaluar (Hu, et. Al, 2020)

   Beneficios de la regresión logística.

   Simplicidad: Los algoritmos de aprendizaje basados en regresión logística son fáciles de comprender e implementar en machine learning. (Fiuza y Rodríguez, 2000).

   Velocidad: Estos algoritmos basados en regresión logística operan los datos mucho más rápidos que otros algoritmos de aprendizaje e incluso en grandes volúmenes del orden de big data. A su vez, necesitan menos capacidad de cómputo, como almacenamiento y potencia de procesamiento, logrando una facilidad de computación en la nube. (Fiuza y Rodríguez, 2000)

   Flexibilidad: Estos algoritmos, permiten encontrar soluciones de más de dos resultados finitos. Son muy usados también en etapas de preprocesamiento de datos como ordenar datos en un rango muy amplio de valores, reduciéndolos a pequeños rangos (Fiuza y Rodríguez, 2000).

Aplicaciones de la regresión logística

La regresión logística tiene varias aplicaciones del mundo real en muchos sectores diferentes.

Fabricación: Las fábricas usan la regresión logística para estimar la probabilidad de fallo de una pieza en la maquinaria. Planifican los programas de mantenimiento en función de las estimaciones y poder minimizar los fallos futuros (Reyes, Escoba, Duarte y Ramirez, 2007)

Sanidad: En el campo de la salud la regresión logística se usa para planificar la atención y el tratamiento preventivo a través de la predicción de la probabilidad de enfermedad en los pacientes. Se utilizan modelos de regresión para comparar la relación de los antecedentes familiares o los genes en las enfermedades (Reyes et al., 2007).

Finanzas: En el ámbito financiero la regresión logística analiza las transacciones en busca de fraudes y evaluación de solicitudes de préstamos (Reyes et al., 2007).

Funcionamiento del análisis de regresión

La regresión logística es una de las técnicas de análisis de regresión más utilizadas que los científicos de datos utilizan constantemente en Machine Learning (ML) (García, et al, 2010)

1.1.- **Identificar la pregunta:**

Para el análisis de regresión logística, hay que formular la pregunta para obtener resultados concretos:

¿El cambio del tiempo de manera brusca afectan nuestras ventas mensuales?

(sí o no)

¿Qué tipo de actividad de tarjeta de crédito realizan nuestros proveedores?

1.2.- **Recopilar datos históricos:**

Identificada la pregunta, se debe identificar las dimensiones o viables que intervienen. Se recopila los datos para todas las dimensiones o variables; es decir

elaborar el set de datos, en el caso del ejemplo anterior, registrar datos de las ventas mensuales realizado en los últimos 6 meses para hacer el entrenamiento del algoritmo.

1.3.- Entrenar el modelo de análisis de regresión:

Consiste en procesar la colección de datos mediante el algoritmo o método de regresión, llamado a este proceso entrenamiento, se procesará diferentes datos y mediante ecuaciones. Ejemplo, si durante 3 meses, se registraron lluvias 7, 9 y 10 días respectivamente y el número de ventas en ese tiempo fue de 12, 15, 25, el algoritmo de regresión conectará los factores con la ecuación:

Ventas = 1.5* (días de lluvia) + 7

1.4.- Probar el modelo, haciendo predicciones con valores nuevos:

Para valores nuevos, se utiliza la ecuación para hacer una predicción. Si sabe que lloverá durante cinco días en julio, la regresión calculará el valor de venta de julio en 14.5

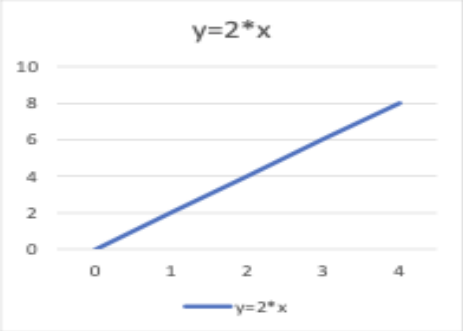
Funcionamiento del modelo de regresión logística

Ecuaciones:
Las ecuaciones matemáticas, establecen la relación entre dos variables: x e y.

Puede usar estas ecuaciones, para trazar un gráfico a lo largo de los ejes x e y indicando diferentes valores de x e y.

Por ejemplo, si traza el gráfico para la función $y = 2*x$, obtendrá una línea recta como se muestra a continuación. Por lo tanto, esta función también se denomina función lineal.

Figura 9
Función lineal



Fuente: (López, s.f.)

Variables

Son propiedades o características de personas u objetos factibles de ser medibles y relevantes para un caso de estudio, ejemplo el género de un estudiante, la edad, apellidos, teléfono, religión, marca de vehículo, modelo etc. Para la investigación las variables son importantes cuando estas se relacionan entre si logran formar parte de una hipótesis o teoría científica. Der así, a estas variables relacionadas se les denomina "constructos o construcciones hipotéticas" (Hernández, Fernández y Baptista, 2006)

Función de regresión logística

La función logística o función logit, es un rutina o algoritmo que cuando es invocado, devuelve un valor asociado con algún parámetro de entrada. Esta función pertenece al método estadístico conocido como regresión logística. Se mapea como una función sigmoidea de *x*.

Figura 10

Ecuación de regresión logística

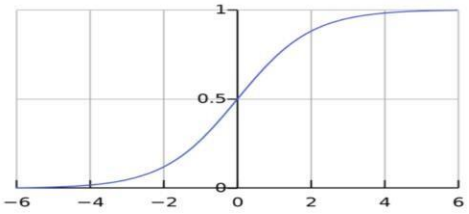
$$f(x) = \frac{1}{1 + e^{-x}} \quad \text{..... Ec (1)}$$

Fuente: López, s.f.

Si grafica esta ecuación de regresión, obtendrá una curva en S como la que se muestra a continuación.

Figura 11

Gráfica de la regresión logística



Fuente:López, s.f.

La función logit como toda función al ser invocada, devuelve un numero entero entre 0 y 1 para la variable de clase o dependiente. Estos valores dependerán de los valores de la variable independiente. (López, s.f.)

Análisis de regresión logística con múltiples variables independientes

Es frecuente que muchas variables independientes afectan al valor de la variable dependiente. Para representar un modelo de esta naturaleza, las fórmulas de regresión logística adoptan una relación lineal entre las diferentes variables independientes (López, s.f.). Se modifica la función sigmoidea y calcula la variable de salida final. Se conoce a esta expresión como función de regresión lineal múltiple.

Figura 12

Ecuación de la regresión logística múltiple

$$Y=f(\beta_0 + \beta_{1x1} + \beta_{2x2} + \ldots \beta_{nxn} +) \ldots\ldots\ldots\ldots\ldots\ldots \text{Ec (2)}$$

Fuente: López, s.f.

Donde β es el coeficiente de la regresión. La función, logit puede calcular de forma inversa estos valores de coeficientes al suministrar un conjunto de datos grande con valores conocidos de variables dependientes e independientes.

Figura 13

Representación de la función logística

$$\text{Logit Function} = \log(p/1-p) \qquad \text{Ec (3)}$$

Fuente: López, s.f.

Tipos de análisis de regresión logística.

a.- Regresión logística binaria

Es utilizada para conocer la relación entre una variable dependiente de tipo cualitativa y una o más variables independientes denominadas explicativas, pudiendo ser cualitativas o cuantitativas, con el fin de obtener una estimación de probabilidad de ocurrir un evento a partir de una o más variables independientes. (Lawton & Brody, citado en García, et al; 2010)

b.- Regresión logística multinomial

Es una generalización del modelo de regresión logística, donde la variable dependiente, tiene más de tres categorías. El algoritmo asume que Y tiene una distribución multinomial (Agresti, 2007 [2] Silva Ay¸ Caguer y Barroso, 2004 (citado por Escalona; 2020).

Este método de regresión multinomial, reporta los resultados con valores comprendidos entre 0 y 1 y como la función logística devuelve valores continuos, tales como: 0.1, 0.11, 0.13, etc., la regresión multinomial, agrupa los resultados a los valores más cercanos posibles a 0 o 1.

2. Árboles de decisión

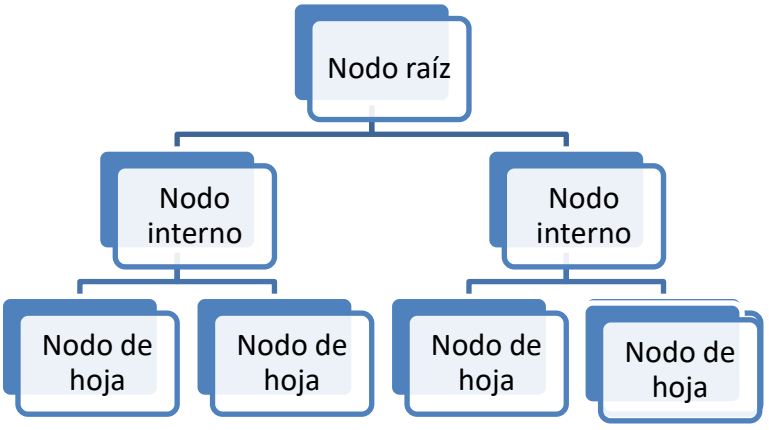
Son una gama de algoritmos de aprendizaje supervisado en Machine Learning usados en modelos de predictivos, tienen por finalidad lograr un aprendizaje de modo inductivo a partir de hechos y expresiones. Se asemejan a los sistemas de predictivos basados en reglas. Representan y categorizan una serie de opciones que aparecen de modo secuencial en la solución de una problemática. El conocimiento generado durante el proceso de aprendizaje inductivo se esquematiza mediante un gráfico de

árbol, donde la raíz o nodo principal es la característica desde la cual se realiza el proceso de la clasificación. Cada respuesta a las preguntas que se vayan haciendo, se representa mediante un nodo hijo.

Las ramas de cada uno de estos nodos se etiquetan con los posibles valores del atributo. (Rusell, et. al; s/f). Los valores de las variables pueden ser discretos o continuos (Breiman, citado por Barrientos, et. al.; 2019)

Figura 14

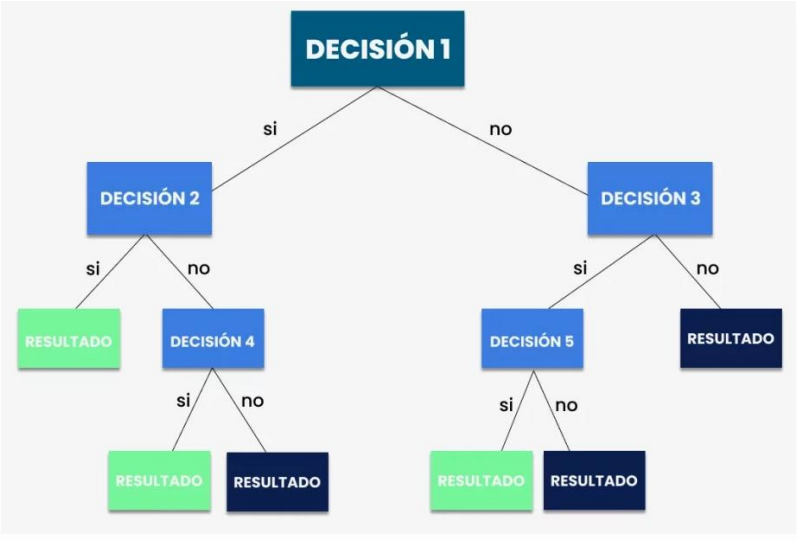
Estructura de un árbol de decisión



Fuente: López, s.f.

Figura 15

Ejemplo de un árbol de decisión



Fuente: Rusell, et. al;s.f.

La estructura crea representaciones sencillas de comprender para situaciones de toma de decisiones, logrando que los integrantes de una organización o equipo de trabajo, entienda por que se tomó tal o cual decisión.

El aprendizaje del árbol de decisiones emplea una estrategia de divide y vencerás mediante la realización de una búsqueda codiciosa para identificar los puntos de división óptimos dentro de un árbol. La división se repite recursivamente de arriba a abajo hasta que los registros se hayan clasificado y eqtiquetados(Rusell, et. al;s.f.).

Tipos de árboles de decisión

Se tiene los siguientes:

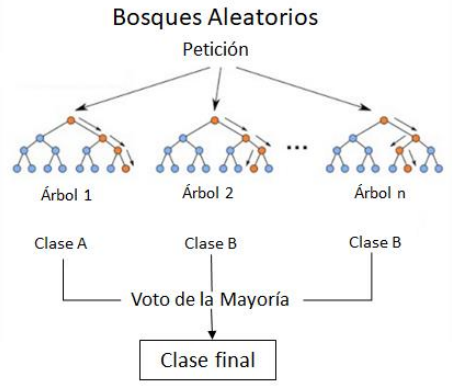
- ID3: Significa "Iterative Dichotomiser 3". Usa la entropía y ganancia de información como indicadores para realizar las divisiones de las hojas.
- C4.5: Usa la ganancia de información o partes de ella, para evaluar los puntos de división en los árboles de decisión.
- CART: Significa "árboles de clasificación y regresion (“classification and regression trees”). Utiliza la impureza de Gini para identificar el mejor atributo para la división. Esta metrica, mide la frecuencia de clasificación incorrectamente un atributo elegido aleatoriamente. Al evaluar la impureza de Gini, un valor más pequeño es el mejor.

3. ***Bosques Aleatorios*** (González, 2019a)

   Es uno de los algoritmos muy utilizados, por su simplicidad y su aplicación en tareas de clasificación comode regresión.

Figura 16

Regresión con bosques aleatorios



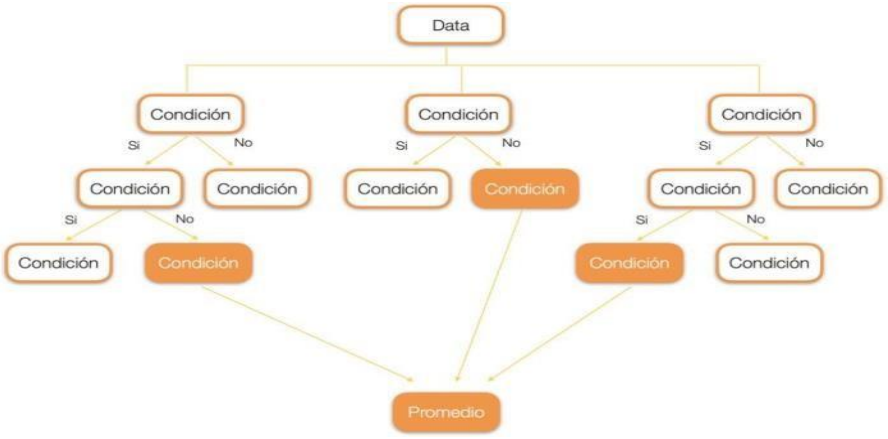
Fuente: (González, 2019a)

Definición

Los Bosques Aleatorios es un algoritmo de aprendizaje supervisado. Como su nombre lo dice, crea bosques de forma aleatorio, compuestos por múltiples árboles de decisión. La cantidad de árboles de cada bosque, puede ser ingresado como parámetro. Combina los árboles para obtener un predicciones más precisas y estables. A más árboles en el bosque, más robusto es el bosque.

Figura 17

Estructura de un Bosque aleatorios

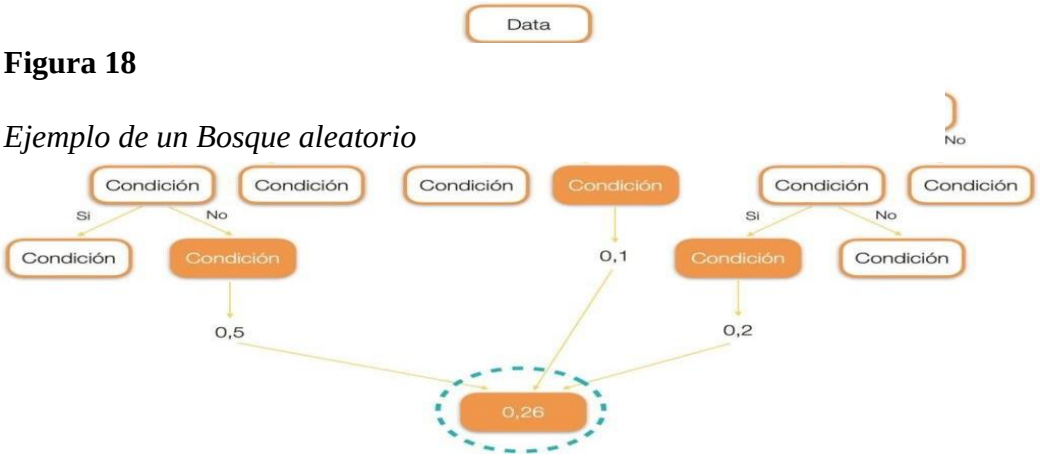


Fuente:
(González, 2019a)

El algoritmo usa aleatoriedad adicional al modelo. A medida que crecen los árboles, busca el mejor atributo entre un conjunto de características al azar, lográndose una diversidad y convirtiendo por tanto en un mejor modelo. En Bosques Aleatorios, el algoritmo para dividir un nodo sólo tiene en cuenta un subconjunto al azar de los atributos, asi los árboles sean más aleatorios, mediante el uso de umbrales en cada función.

Figura 18

Ejemplo de un Bosque aleatorio



Fuente: (González, 2019a)

4. Máquinas de Vectores de Soporte

Los algoritmos de aprendizaje supervisados denominados máquinas de vectores de soporte, fuero fue desarrollado en los laboratorios de AT&T Bell por un equipo de investigadores liderado por Vladimir Vapnik (Vapnik,1995).

Según (Burges, 1998).

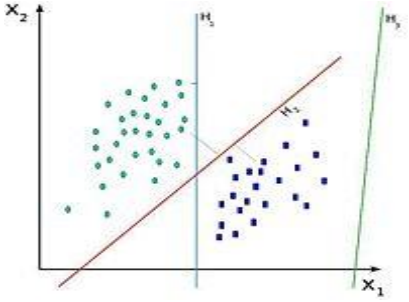
A partir de un conjunto de datos, se etiqueta las clases y elige una instancia del algoritmo de SVM para entrenar el conjunto de datos y construir el modelo para predecir la clase de una nueva muestra. Un SVM grafica un conjunto de puntos en el espacio y las clases son separadas en dos grupos los mas distantes posibles mediante un hiperplano de separación que se define como el vector entre los 2 puntos, de las 2 clases, más cercanos al que se llama vector soporte.

(Cortes, Vapnik, 1995), Una SVM, produce un conjunto de hiperplanos en un entorno de dimensionalidad grande. Una clasificación optima cuando un hiperplano que separa la clase lo mayor posible independientes posibles; es decir mayor distancia de separación.

Justamente en el concepto de separación optima, es lo que caracteriza las SVM; es decir el algoritmo SVM, busca un hiperplano que, al separar las clases, ubica a los puntos de cada clase mas cerca al hiperplano y entre ellos elige los que estén más alejados del vector. A estos algoritmos también se le conoce como clasificadores de margen máximo.

Por ejemplo, en el presente gráfico, los datos se encuentran en el plano cartesiano x-y. El algoritmo de aprendizaje SVM, encuentra un vector que asocia a las variables predictoras y define el límite de un input y verifica si corresponde a una categoría u otra. Podrían existir un número infinito de posibles hiperplanos, que realicen la clasificación.

Figura 19 Hiperplanos generados

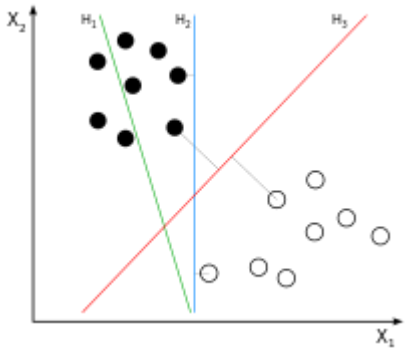


Fuente: Burges, 1998

¿Cuál es la mejor y cómo la definimos?

Figura 20

H3 como mejor recta de clasificación



Fuente: González, 2019b

En el ejemplo, H1 separa las clases erróneamente. H2 separa las clases mejor, con un margen ajustado. H3 las separa con el margen máximo, por tanto, es el óptimo.

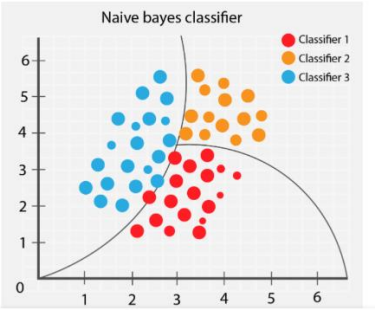
Se les llama vectores de soporte a los puntos que conforman las dos líneas paralelas al hiperplano, donde el espacio entre ellas, respecto del margen sea la máxima posible.

5. ***Naive Bayes*** *(González, 2019b)*

   *Naive Bayes o el Ingenuo Bayes, es uno de lo*s algoritmos más simples y poderosos para la clasificación basado en el Teorema de Bayes, con una suposición de independencia entre los predictores. Naive Bayes es fácil de construir y particularmente útil para conjuntos de datos muy grandes.

Figura 21

Representa una regresión con Naive Bayes



Fuente: Gonzalez, 2019b)

Definición

El clasificador Naive Bayes considera que el efecto de una característica particular en una clase es independiente de otras características. Por ejemplo, un prestamista es deseable o no dependiendo de sus ingresos, historial de préstamos y transacciones anteriores, edad y ubicación. Si estas características son interdependientes, son consideradas de forma independiente. La suposición simplifica la computación, y por eso se considera ingenua. A Esto se denomina independencia condicional de clase.

Representación matemática de Naive Bayes

La fórmula del teorema de Bayes es la siguiente:

$$P(h/D) = P(D/h) * P(h) / P(D) \quad \text{..... Ec (8)}$$

Donde:

P(h): es la probabilidad de que la hipótesis h sea cierta (independientemente de los datos). Esto se conoce como la probabilidad a priori de h.

P(D): probabilidad de los datos (independientemente de la hipótesis). Esto se conoce como probabilidad previa.

P(h|D): es la probabilidad de la hipótesis h dada los datos D. Esto se conoce como la probabilidad posterior.

P(D|h): es la probabilidad de los datos d, dado que la hipótesis h era cierta. Esto se conoce como probabilidad de verosimilitud.

En caso de que se tenga una sola característica, el clasificador Naive Bayes calcula la probabilidad de un evento en los siguientes pasos:

Paso 1: calcular la probabilidad previa para las etiquetas de clase dadas.
Paso 2: determinar la probabilidad de probabilidad con cada atributo para cadaclase.
Paso 3: poner estos valores en el teorema de Bayes y calcular la probabilidad posterior.

Paso 4: ver qué clase tiene una probabilidad más alta, dado que la variable de entrada pertenece a la clase de probabilidad más alta.

Naive Bayes es el algoritmo más sencillo y potente. A pesar de los significativos avances de Machine Learning en los últimos años, ha demostrado su valía. Se ha implementado con éxito en muchas aplicaciones, desde el análisis de texto hasta los motores de recomendación.

METODOLOGÍAS UTILIZADAS EN MODELOS PREDICTIVOS

1. El Proceso KDD

Según Calvache-Fernández, L. C., Álvarez-Vallejo, V., & Triviño-Arbeláez, J. I. (2018). El proceso KDD es utilizado para extraer patrones de comportamiento de forma automatizada a partir de grandes volúmenes de información del orden del big data. Es un proceso iterativo y se aplica tantas hasta obtener la información necesaria. KDD, tiene como motivación la detección de información que permita resolver los problemas o necesidades que surgen en las empresas y es a menudo solicitado por directivos y/o stakeholders.

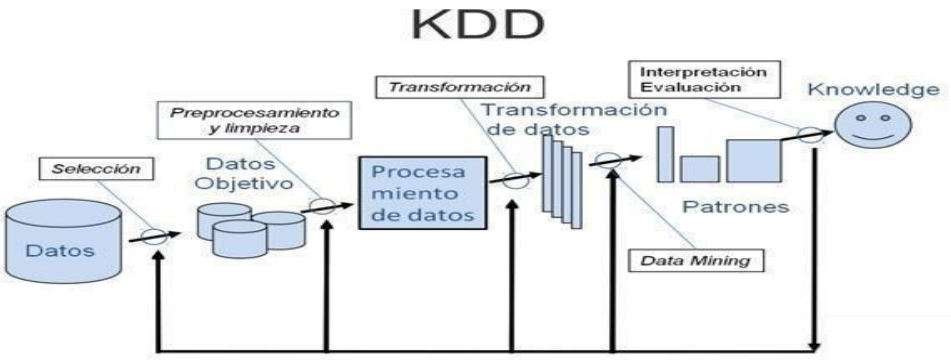
El conocimiento que se pretende extraer con el proceso KDD debe ser no trivial,implícito, previamente desconocido y potencialmente útil.

Se consideran cinco etapas:

1. **Selección:** Se define el objetivo del modelo e identifican las variables que conformaran el DataSet.
2. **Procesamiento previo:** Se realiza la limpieza d ellos datos, buscando valores perdidos o nulos y define los criterios de solución.
3. **Transformación:** Esta etapa en la discretización de la data y resolver el problema de la dimensionalidad.
4. **Minería de Datos:** Usando métodos o algoritmos de aprendizaje, se dedica a la búsqueda de patrones comunes de una forma representacional de acuerdo a los objetivos de la minería de datos.
5. **Interpretación / Evaluación**: Se evalúa e interpreta los patrones obtenidos por el modelo.

Figura 22

Etapas en el procedimiento KDD



Nota: Fuente: Fayyad et al. (1995).

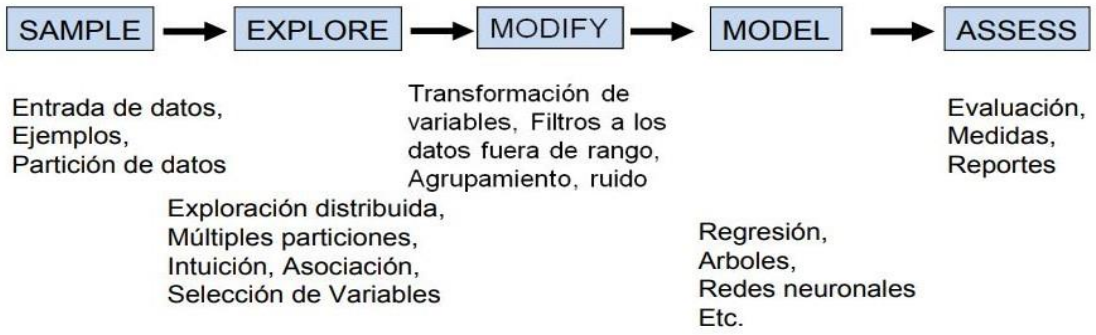
2. El proceso SEMMA

El acrónimo SEMMA surge de las iniciales de las palabras Sample (muestra), Explore (explorar), Modify (modificar),Model (modelar) y Assess (evaluar).

Según Camargo, Silva (2010), indicaron que SEMMA es una organización para el manejo de una herramienta funcional de SAS llamada Enterprise Manager para el manejo de tareas de minería de datos. También indicaron que SEMMA intenta hacer fácil de aplicar la exploración estadística y la visualización de técnicas, seleccionando y transformando las variables predictivas más relevantes, modelándolas para obtener resultados, y finalmente confirmar la precisión del modelo.

Figura 23

Fases y actividades del proceso SEMMA



Fuente: Fayyad et al. (1995).

1. **Muestra:** Define el set de datos los más grande posible para poder encontrar mejores patrones en cantidad, pero pequeño en dimensionalidad.
2. **Explorar:** Consiste en la exploración para determinar la limpieza de los datos, ubicando valores perdidos o nulos y tratarlos mediante criterios establecidos.
3. **Modificar:** Consiste en la discretización de datos y atacando el problema de la dimensionalidad a fin de obtener un set de datos definitivo.
4. **Modelo:** aplicación de algún método de entrenamiento al set de datos elegido a fin de encontrar el modelo propuesto y validarlo.
5. **Evaluar:** Se realiza la evaluación del modelo con datos de prueba a fin de verificar su confiabilidad.

El proceso SEMMA, pretende ser una guía para el usuario en la implementación de proyectos de minería de datos, ofreciendo un proceso de fácil comprensión, desarrollo y mantenimiento adecuado.

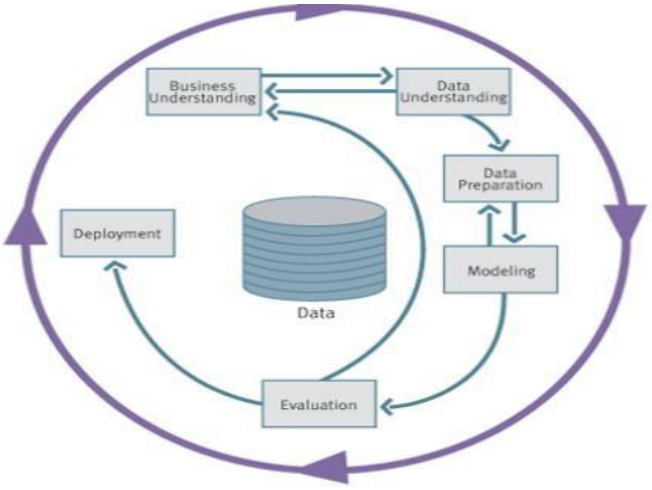
3. **Metodología CRISP-DM** (Girones, 2017)

El proceso CRISP-DM se desarrolló por medio del esfuerzo de un consorcio inicialmentecompuesto por Daimler Chryrler, SPSS y NCR. CRISP-DM (CROSS-Industry StandardProcess for Data Mining).

Consiste en un ciclo que consta de seis etapas:

Figura 24

Fases de la metodología CRISP-DM



Fuente: (Girones, 2017)

1. **Comprensión del negocio**: Define el objetivo del proyecto y conocimiento del negocio.
2. **Comprensión de los datos:** Recopilación de los datos, selecciona los atributos más relevantes para el set de datos de la investigación e implementa el set de datos base.
3. **Preparación de datos:** En esta etapa se realiza la discretización de datos, se trata la dimensionalidad de la data y limpieza de datos encontrando valores perdidos o erróneos y establece criterios para su solución.
4. **Modelado:** En esta fase, se seleccionan los algoritmos de Machine Learning a utilizarse en el entrenamiento aplicado al set de datos y obtener los modelos. Se validan sus parámetros a valores óptimos.
5. **Evaluación:** Los modelos obtenidos se evalúan para determinar su grado de confiabilidad a través de diversas métricas tales como coeficiente de determinación o matriz de confusión.
6. **Implementación:** El modelo final, no es el producto final de un proyecto de minería de datos. El modelo validado, probado y propuesto, debe pasar a producción para la implementación en un sistema web, app o aplicación de escritorio para uso de los usuarios finales.

COMPARATIVO ENTRE METODOLOGÍAS

1.- Comparación KDD y SEMMA:

Ambos procesos son equivalentes, las etapas del proceso SEMMA, vendrían a ser la implementación practica de las 5 etapas del proceso KDD, dado que esta última está vinculada al software SAS Enterprise Miner, tal como se observa en la tabla 3.

Tabla 3

Comparación KDD - SEMMA

KDD	SEMMA
Muestra	Selección
Explorar	Pre-procesamiento
Modificar	Transformación
Modelo	Data Mining
Evaluación	Interpretación/Evaluación

Si comparamos KDD y CRISP-DM podemos observar que la metodología CRISP-DM incorpora los pasos que deben preceder y seguir el proceso KDD, como se observa en la tabla 4.

Tabla 4

Comparación KDD – CRISP - DM

KDD	CRISP-DM
La fase de Entendimientode Negocios	Puede identificarse con el desarrollo de una comprensión del dominio de la aplicación, el conocimiento previo relevante y los objetivos del usuario final.
La fase de implementación	Puede identificarse con la consolidación incorporando este conocimiento en el sistema.
La fase de Entendimiento de Datos	Puede ser identificada como la combinación de Selección y Pre procesamiento.
La fase de preparación	Se identifica con discretización o tratamiento de dimensionalidad.
La fase de modelado	Se identifica con minería de datos
La fase de Evaluación	Se identifica con evaluación.

2.- Comparación CRISP –DM Y SEMMA

Tabla 5

Comparación SEMA – CRISP – DM

SEMA	CRISP-DM
Muestra y Exploración	Conocimiento y comprensión de Datos
Modificar	Preparación de datos, discretización y dimensionalidad.
Modelo	Entrenamiento y modelado,
Evaluar paralelos	Evaluación del modelo.

Nota: Ambos modelos pretenden ser algo cíclicos en lugar de lineales.

3.- Diferencias entre CRISP-DM y SEMMA

SEMMA se desarrolló para un software especifico de software: SAS Enterprise Miner

E incide menos énfasis en las fases de planificación inicial consideradas por CRISP-DM (Comprensión empresarial y comprensión de datos). Omite totalmente la fase de implementación.

ELECCIÓN DE LA METODOLOGÍA

Deseamos una metodología que sea de amplio uso entre profesionales por lo cual recurrimos a un análisis del portal kdnuggets en el que se puede ver la preferencia de 200 usuarios en el uso de una metodología de minería de datos.

Las encuestas realizadas por kdnuggets entre 2007 y 2014 tienden a reflejar el dominio de CRISP-DM como metodología más empleada en la gestión de proyectos de Data Mining, seguida de SEMMA y KDD, no obstante, existe un porcentaje importante de expertos que emplea su propia metodología, una establecida por la organización en la que trabaja u otras metodologías de menor uso.

Vemos que en primer lugar de preferencia se encuentra la metodología CRISP –DM con un 43%, en segundo lugar, SEMMA con un 8.5% y en tercer lugar KDD con 7.5%.

Tabla 6

La distribución regional de los votantes

¿Qué metodología principal está utilizando para sus proyectos de análisis, minería de datos o ciencia de datos? [200 votos en total]		
	Encuesta 2014	Encuesta 2007
CRISP – DM (86)	43%	42%
Mi propio (55)	27.5%	19%
SEMMA (17)	8.5%	13%
Otros, no específicos del dominio (16)	8%	4%
Proceso KDD (15)	7.5%	7.3%
Mis organizaciones' (7)	3.5%	5.3%
Una metodología específica de dominio (4)	2%	4.7%
Ninguna (0)	0%	4.7%

La distribución regional de los votantes fueEE. UU./Canadá, 45,5 %

Europa, 28,5%

Asia, 14%

América Latina, 9,5%

Otros, 2,5%

Estas tres metodologías han sido comparadas en numerosos artículos científicos por la compañía kdnuggets y suelen ser las metodologías de uso genérico más utilizadas por los expertos, tal como suelen revelar las encuestas publicadas en el sitio web kdnuggets,uno de los mayores puntos de encuentro en internet entre expertos en la materia.

2.2. Marco conceptual

El marco conceptual es una investigación bibliográfica que habla de las variables que se estudiarán en la investigación, o de la relación existente entre ellas, descritas en estudios semejantes o previos. “Hace referencia a perspectivas o enfoques teóricos empleados en estudios relacionados, se analiza su bondad o propiedad.” (López EK, Juárez F, Acevedo M. 2010).

Modelo

Según, el Diccionario de la lengua española (Real Academia Española, 2019), un modelo es un o punto de referencia para imitar o reproducir. Se dice también que es la representación pequeña de alguna cosa o también un esquema teórico o forma matemática, de un sistema o de una realidad compleja, como el crecimiento y desarrollo de un país.

2.2.1. Modelamiento predictivo

Cuando hablamos de predicción de modo global, se entiende como pronosticar el futuro o ahondar en lo desconocido, esto se entendió así desde mucho antes que el método científico apareciera, quienes realizaban estas actividades eran los astrólogos o los brujos.

“ ..el modelamiento predictivo o análisis predictivo, indicaremos que este se realiza empleando una serie de técnicas estadísticas y computacionales de manera ordenada para pronosticar resultados futuros a partir de los datos del pasado” (Cano, Berea, 2017).

2.2.2. La minería de datos:

La minería de datos es un proceso automático en el que se combinan

descubrimiento y análisis. El proceso consiste en extraer patrones en forma de reglas o funciones, a partir de los datos, para que el usuario los analice. Esta tarea implica generalmente pre procesar los datos, hacer minería de datos y evaluar e interpretar los resultados (Pereira, R. T., 2010).

El proceso de esta técnica es iterativo e interactivo. Iterativo porque las etapas pueden ser retroalimentadas e interactivo porque el usuario interactúa muchas veces con el modelo para la toma de decisiones.

2.2.3. Desertor

Persona que ha abandonado los estudios y sus obligaciones de estudiante y por tanto su calidad de alumno y derechos adquiridos en su inscripción en el centro educativo.

En el presente artículo, (Comisión Intersectorial de Reinserción Educativa, 2006). la deserción es comprendida como un proceso de alejamiento y de abandono paulatino de un espacio cotidiano (como es la escuela) que implica también el abandono de ciertos ritos personales y familiares que inciden en el desarrollo de la identidad y la proyección personal de un niño.

2.2.4. Desempeño académico

(Salvador, García, Valcárcel, Muñoz & Repiso, 1989), es aquel que puede ser evaluado al finalizar los estudios con las tasas de promoción, la repetición de la misma asignatura y el abandono cuando dejan de matricularse en las asignaturas de la carrera.

2.2.5. Ajuste de Hiper parámetros

Cuando se entrenan modelos de Machine Learning, cada conjunto de datos y los modelos resultantes, necesitan de un conjunto diferente de hiper parámetros, que son un tipo de variables. Se determinan mediante la realización de múltiples experimentos, en los que se elige un conjunto de hiper parámetros y se los ejecuta a través del modelo.

Las metaheurísticas son unos de los algoritmos actuales más populares para la resolución de problemas de optimización. Se inspiran en la simulación de fenómenos naturales, comportamientos sociales de especies, o el proceso evolutivo, obteniendo soluciones a problemas donde los métodos exactos y las heurísticas fallan al quedar atrapados en óptimos locales o ser demasiado costosos computacionalmente. Cada metaheurística tiene un comportamiento distinto que depende de unos hiper parámetros que son ajustados manualmente para la resolución eficaz de un problema específico. Sin embargo, la selección de valores adecuados para estos hiper parámetros es una tarea compleja y una de las principales vías de investigación en el campo de las metaheurísticas (Domínguez, 2021).

2.2.6. Anaconda

Es una distribución de Python para el procesamiento de datos a gran escala, el análisis predictivo y la computación científica. Anaconda incluye NumPy, SciPy, matplotlib, pandas, IPython, Jupyter Notebook y scikit-learn (Müller & Guido, 2016).

2.2.7. Jupyter Notebook

Entorno interactivo para ejecutar código en el navegador, es una herramienta para el análisis de datos exploratorios y es ampliamente utilizada por los científicos de datos. Jupyter Notebook acepta muchos lenguajes de programación, pero basta el soporte de Python, al mismo tiempo facilita la incorporación de código, texto e imágenes (Müller & Guido, 2016).

2.2.8. Numpy

Python, es un paquete fundamental para la computación científica porque trabaja con matrices multidimensionales, funciones matemáticas de alto nivel, tales como operaciones de álgebra lineal, transformada de Fourier, y generadores de números pseudo aleatorios (Müller & Guido, 2016).

2.2.9. **Pandas**

Es una biblioteca de Python para gestión y análisis de datos. Pandas, trabaja con una estructura de datos llamada DataFrame que es una tabla, similar a una hoja de cálculo de Excel. Pandas proporciona una gran variedad de métodos para modificar y operar en esta tabla; permite consultas de tipo SQL y uniones de tablas. A diferencia de NumPy, que requiere que todas las entradas en una matriz sean del mismo tipo, pandas permite que cada columna tenga un tipo separado (por ejemplo, números enteros, fechas, números de punto flotante y cadenas). Otra herramienta valiosa proporcionada por pandas es su capacidad para ingerir desde una gran variedad de formatos de archivo y bases de datos, como SQL, archivos de Excel y archivos de valores separados por comas (CSV) (Müller & Guido, 2016).

2.2.10. **Scikit-Learn,** es una de estas librerías gratuitas para Python. Cuenta con algoritmosde clasificación, regresión, clustering y reducción de dimensionalidad. Además, presenta la compatibilidad con otras librerías de Python como NumPy, SciPy y matplotlib (Müller & Guido, 2016).

CAPÍTULO III MARCO METODOLÓGICO

3.1. Hipótesis central de la investigación

Un modelo predictivo basado en algoritmos de aprendizaje supervisado, permitirá predecir la deserción estudiantil con un grado de confianza superior a 80% en estudiantes que cursan el primer año académico de los centros de Educación Superior Tecnológicos Públicos de la región La Libertad.

3.2. Variables e indicadores de la investigación

3.2.1 Variables

1 Definición conceptual

V1: Modelo predictivo, basado en algoritmos de aprendizaje automático supervisados

V2: Deserción estudiantil, estima el grado de deserción estudiantil en los institutos superiores tecnológicos de la región La Libertad.

2 Definición operacional

V1: Modelo predictivo, basado en el entrenamiento de los algoritmos para obtener los modelos de entrada del modelo propuesto.

V2: Deserción estudiantil, a través del coeficiente de correlación de pearson se pudo determinar la dimensionalidad del juego de datos final que servirá para entrenar los algoritmos.

3.2.2 Indicadores

Tabla 7

Indicadores

Variables		Dimensiones	Indicadores
Mejor Modelo predictivo	Regresión logística binaria	Tiempo	Semestre
		Desertores	Cantidad por periodo
	Modelo Naive Bayes	Tiempo	Semestre
		Desertores	Cantidad por periodo
	Bosques aleatorios	Tiempo	Semestre
		Desertores	Cantidad por periodo
	Clasificador de Soporte Vectorial	Tiempo	Semestre
		Desertores	Cantidad por periodo
Deserción estudiantil		Desertores	Información de RT
		Carga Familiar	Ficha socioeconómica
		Edad de Estudiante	Ficha socioeconómica
		Situación Laboral	Ficha socioeconómica
		Ingreso Familiar	Ficha Socioeconómica
		Promedio Ponderado	Información de RT
		Vivienda Propia?	Ficha Socioeconómica
		Procedencia (cerca a institución)	Ficha Socioeconómica

3.3. Método de la Investigación

1. Revisión de bibliografía relevante, así como los documentos de gestión institucional.
2. Selección de la muestra de estudio.
3. Selección de características importantes para el desarrollo del modelo.
4. Recolección y limpieza de los datos recabados. Fichas de datos de estudiantes, documentos históricos.
5. Selección de los algoritmos de aprendizaje más utilizados para estos modelos depredicción.

6. Selección de la data para entrenamiento y prueba.
7. Implementación de los algoritmos de aprendizaje.
8. Entrenamiento de los modelos.
9. Determinar la confiabilidad de cada modelo entrenado.
10. Elaboración de una predicción y cotejar con el set de prueba, verificando la confiabilidad de la predicción respecto a la prueba.
11. Resultados y discusión
12. Conclusiones recomendaciones

3.4. Diseño de la investigación

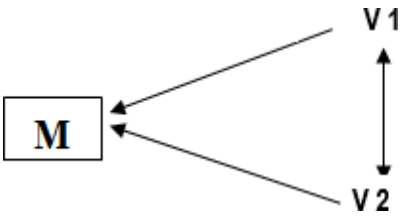
Como el objetivo del presente trabajo de investigación es proponer un modelo predictivo, ensamblado a partir de modelos clásicos y obtener un modelo con buen **índice de confiabilidad para estimar la deserción estudiantil en los centros de Educación Superior Tecnológicos Públicos de la región La Libertad**, basado en máquinas de aprendizaje supervisadas, es que se implementarán varios modelos con los algoritmos de aprendizaje supervisados Regresión logística binaria, Máquinas de Soporte Vectorial, Naive Bayes y Random Forest. Luego se evaluará la confiabilidad de cada modelo y optamos por generar un nuevo modelo ensamblado a partir de los anteriores validados mayor grado de confianza para la propuesta del problema. Los modelos básicos, serán entrenados con el 80% de la data, conformada por información histórica de deserción de 10 años de antigüedad y el 20% restante será para las pruebas de los modelos. Se usarán como datos históricos las fichas de datos de los estudiantes matriculados en los últimos 10 años.

Por lo tanto; en este estudio recurriremos a un ***diseño de investigación con enfoque cuantitativo, no experimental y utilizará el tipo de investigación transversal por ser los datos recogidos en un solo momento*** para el entrenamiento y puesta a prueba del modelo predictivo.

La muestra para la prueba de la hipótesis, será los datos de estudiantes del IESTP Trujillo, provincia Trujillo y departamento la libertad.

Figura 25

Diseño de investigación



V1: Modelo de predictivo

V2: Deserción estudiantil

Respecto al tipo de Investigación, será del tipo ***Aplicada.***

Como no manipularemos variables, la investigación será no experimental**.** "La investigación no experimental es aquella que se realiza sin manipular deliberadamente variables. Es decir, es investigación donde no hacemos variar intencionalmente las variables independientes. Lo que hacemos en la investigación no experimental es observar fenómenos tal y como se dan en su contexto natural, para después analizarlos(Agudelo Viana,2008).

Como pretendemos a través del modelo caracterizar un hecho concreto de la vida real como es la deserción estudiantil esta ***investigación será descriptiva.***

"Consiste, fundamentalmente, en caracterizar un fenómeno o situación concreta indicando sus rasgos más peculiares o diferenciadores. …. (Hernández,Fernández y Baptista, 2006)

En vista que los modelos a estudiar reportan datos numéricos para poder hacer la optimización del modelo propuesto, la investigación tiene un enfoque cuantitativo, "Este enfoque cuantitativo trabaja sobre la base de una revisión de literatura que apunta al tema y da como conclusión un marco teórico orientador de la investigación. Estas recolecciones de datos derivan las hipótesis que serán sometidas a prueba para probarla veracidad del estudio …. (Hernández,Fernández y Baptista, 2006).

Los datos serán recogidos durante una sola observación por lo tanto ***la* investigación será** d***e corte transversal.***

"Los diseños de investigación transeccional o transversal recolectan datos en un solo momento, en un tiempo único. Su propósito es describir variables, y analizar su incidencia o interrelación en un momento dado. Es como tomar una fotografía de un hecho. (Agudelo Viana,2008).

3.5. Población y Muestra

En vista que los institutos superiores tecnológicos públicos de la región, poseen características semejantes entre ellos: periodos académicos, programas de estudios, fechas de admisión, sin costo de estudios, docentes contratados o nombrados por el estado etc. Las problemáticas también son muy semejantes, por esta razón usaremos como población a los estudiantes del IESTP Trujillo de la región la Libertad.

La muestra para el presente proyecto será elegida por el método denominado POR CONVENIENCIA y tomaremos como referencia a los estudiantes del programa: Computación e Informática.

3.6. Técnicas e Instrumentos de Recolección de Datos

1. Análisis documental de los documentos de gestión institucional del IESTP Trujillo.
2. Fichas socioeconómicas de los estudiantes del programa académico de Computación e Informática.
3. Datos de estudiantes egresados en los últimos 10 periodos académicos.

3.7. Procedimiento de la recolección de Datos.

Se recolectaron calificaciones de los primeros 10 últimos periodos académicos del área deregistro técnico del Iestp Trujillo.

Los datos de las fichas socioeconómicas de los estudiantes se recogen del área de bienestar con la confidencialidad respectiva para uso de investigación.

3.8. Técnicas de Procesamiento y análisis de Resultados

Para procesar y analizar los resultados de nuestros modelos, es necesario establecer 2 grupos de datos de nuestro set: Un grupo para entrenar el modelo y un grupo para la validación. Por lo general se divide haciendo un grupo de 80% para el entrenamiento y un 20% para las pruebas. Usaremos para ello el framework anaconda, el cuaderno júpiter notebook como editor de código y python como lenguaje de programación y las librerías: pandas, numpy, scikit Learn y matplotlib.

3.9. Colección de datos caso estudio

El elemento esencial, es el DataSet, para poder abordar un modelo predictivo con machine Learning.

Mostramos cómo se obtuvo y transformó la data para poder generar los conjuntos de datos (DataSet), que nos permitan entrenar, testear y validar el modelo de predicción propuesto en la investigación presente.

a) Data de fichas socioeconómicas

La data de las fichas socioeconómicas obtenida del área de bienestar social se encontróen formato xls (formato de excel) y muchos de ellos en físico.

b) Los promedios ponderados de los primeros periodos académicos

Los datos acerca del promedio del fin del periodo académico, los pudimos obtener del departamento de registro técnico. También estuvo en formato XLS; es decir el formato Excel.

Toda la data tuvimos que unirla a través de un solo DataSet con las características más relevantes y usar el formato XLSX. inicialmente se tuvo 12 características, luego de un análisis de correlación basados en los índices de Pearson, se determinó 7 características que tuvieron mayor significancia respecto a la variable predictora Y.

c) Análisis de los datos

Teniendo el set de datos con las 13 características definidas, se procedió al análisis exploratorio.

Como en muchos casos, la data que se recolecta no siempre se encuentra limpia, encontrándose muchos valores nulos, la estructura de las tablas mal diseñadas, datos de diferente tipo etc. Los códigos de los registros tienen diferente formato y tipos de datos diferentes en la medida que los datos originales vienen de fuentes distintas. Se tuvo que corregir dichas imperfecciones.

Se uso Python, y la librería panda, para encontrar la relación que existe entre cada una de las dimensiones y la variable predictora (Desertor) y fundamentar que las características elegidas son las más relevantes para implementar el modelo predictivo. Usamos también matplotlib y Júpiter Notebook de la suite Anaconda, para representar la información mediante gráficos de calor.

d) Explorando los datos

1. Analizamos la relación existente entre las características y la variable Desertor. Podemos verificar que, si se relacionan, algunas en mayor grado que otras.

2. Mostramos a través de un gráfico de calor los grados de correlación existentes entre todas las características y la variable desertor para una mejor exploración.

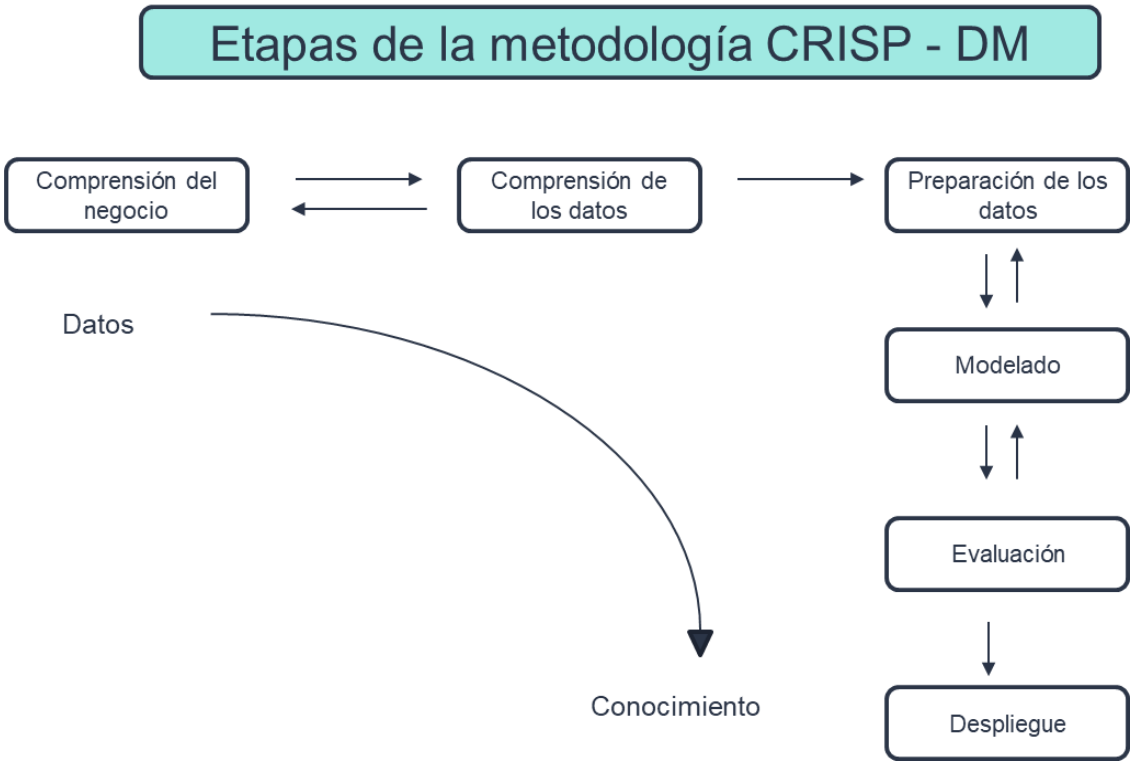
CAPÍTULO IV.- RESULTADOS Y DISCUSIÓN

4.1 RESULTADOS

Metodología usada como guía, CRISP DM

Figura 26

Metodología CRISP DM



4.1.1 Comprensión del negocio:

Figura 27

Comprensión del negocio

a.- Evaluar Situación Actual

La deserción estudiantil en educación superior tecnológica es una problemática que se da en todos los centros de educación superior tecnológicos públicos de la región y el país: la problemática es, como estimar la deserción estudiantil a tiempo y tomar medidas correctivas.

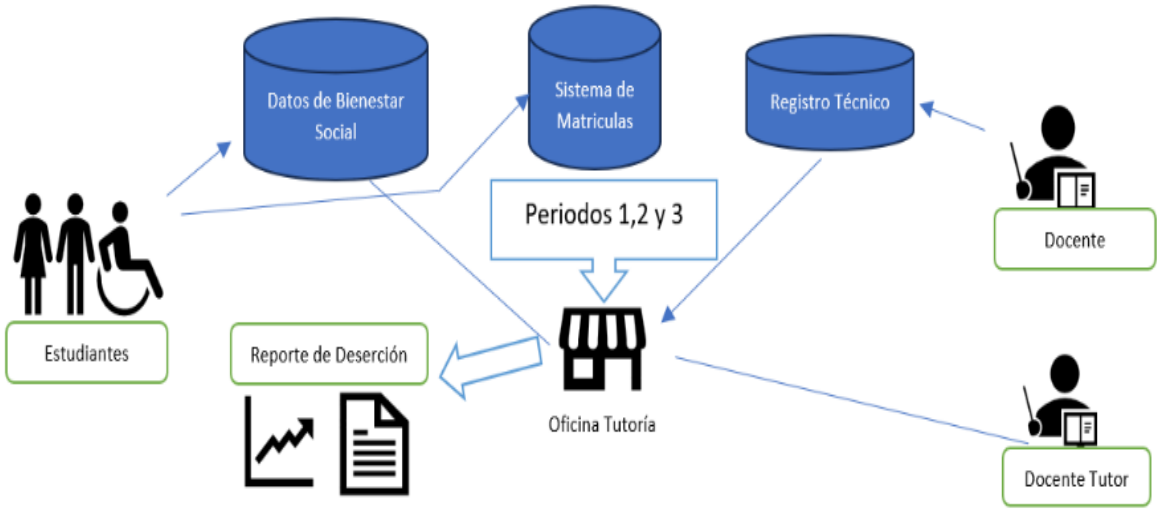
El promedio de estudiantes que concluyen sus estudios oscila entre 40 y 45 % del total de ingresantes y de allí los que llegaban a titularse, están entre el 25 y 30 % de los que terminaron sus estudios en promedio.

La situación es realmente seria y esta problemática se ve en las mismas proporciones en los institutos tecnológicos de la región.

Las cifras de deserción, motivaron a emprender esta investigación para predecir con cierto grado de confiabilidad, estudiantes que podrían desertar al final del primer periodo académico, a fin de que el área de tutoría pueda hacer un seguimiento y mitigar esta problemática. El data está basada en los 10 últimos periodos académicos de estudiantes del programa Computación e Informática.

Figura 28

Comprensión del negocio



b.- Objetivos del negocio

- Lograr la permanencia de los estudiantes en los diferentes programas académicos durante los 3 años de duración de los programas académicos.
- Brindar una asesoría personalizada a los estudiantes con riesgo de deserción.
- Cumplir con los estándares de calidad estipulados por el SINEASE para la acreditación de los programas académicos que se imparte.

c.- Objetivos a nivel de minería de datos

- Establecer un modelo que permita predecir estudiantes que abandonarían los estudios durante los periodos académicos que consta sus programas.
- Encontrar patrones de características de estudiantes con probabilidades de desertar de sus programas de estudios.

4.1.2 Comprensión de los datos:

Se consideraron datos de campo, basados en las fichas de matrículas de alumnos y los reportes de matrículas, notas de los periodos académicos y fichas socio económicas del programa computación e Informática del Iestp Trujillo desde el año 2010, considerando una totalidad de 10 periodos y una data aproximada de 500 registros. Se extraen las características más relevantes, según enfoques teóricos y se construye el Set de Datos primario en una hoja de cálculo con los datos de registros encontrados.

DataSet_Original_Tesis24.xlsx.

Figura 29

Comprensión de los datos

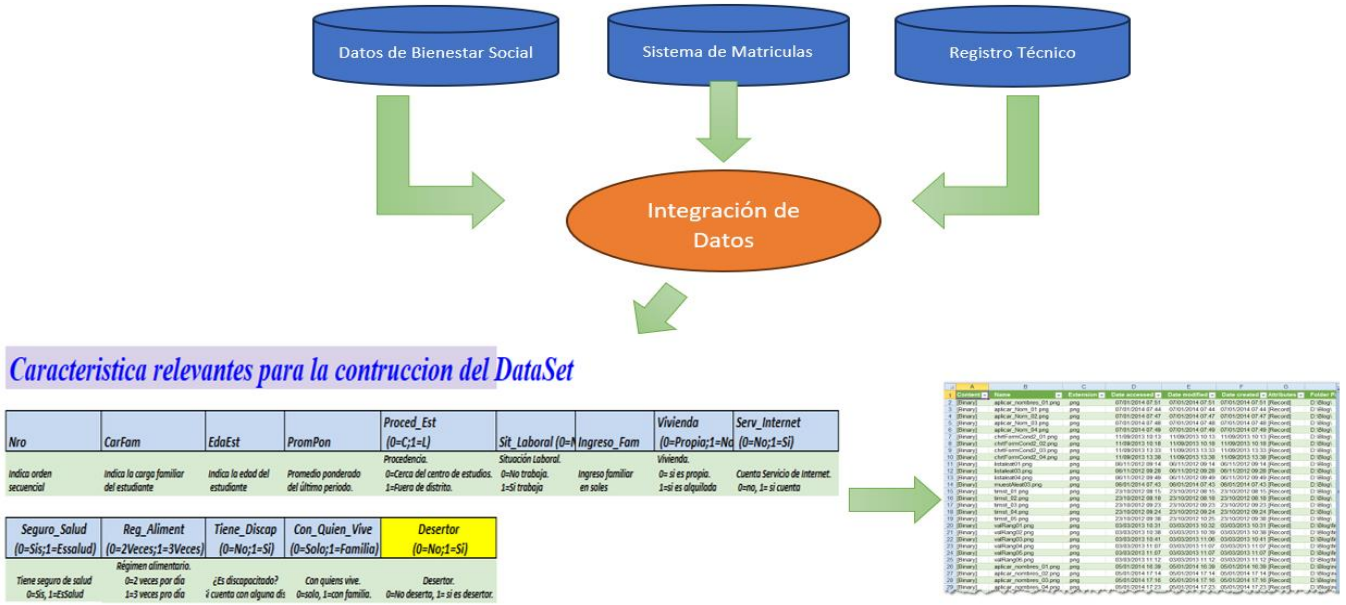


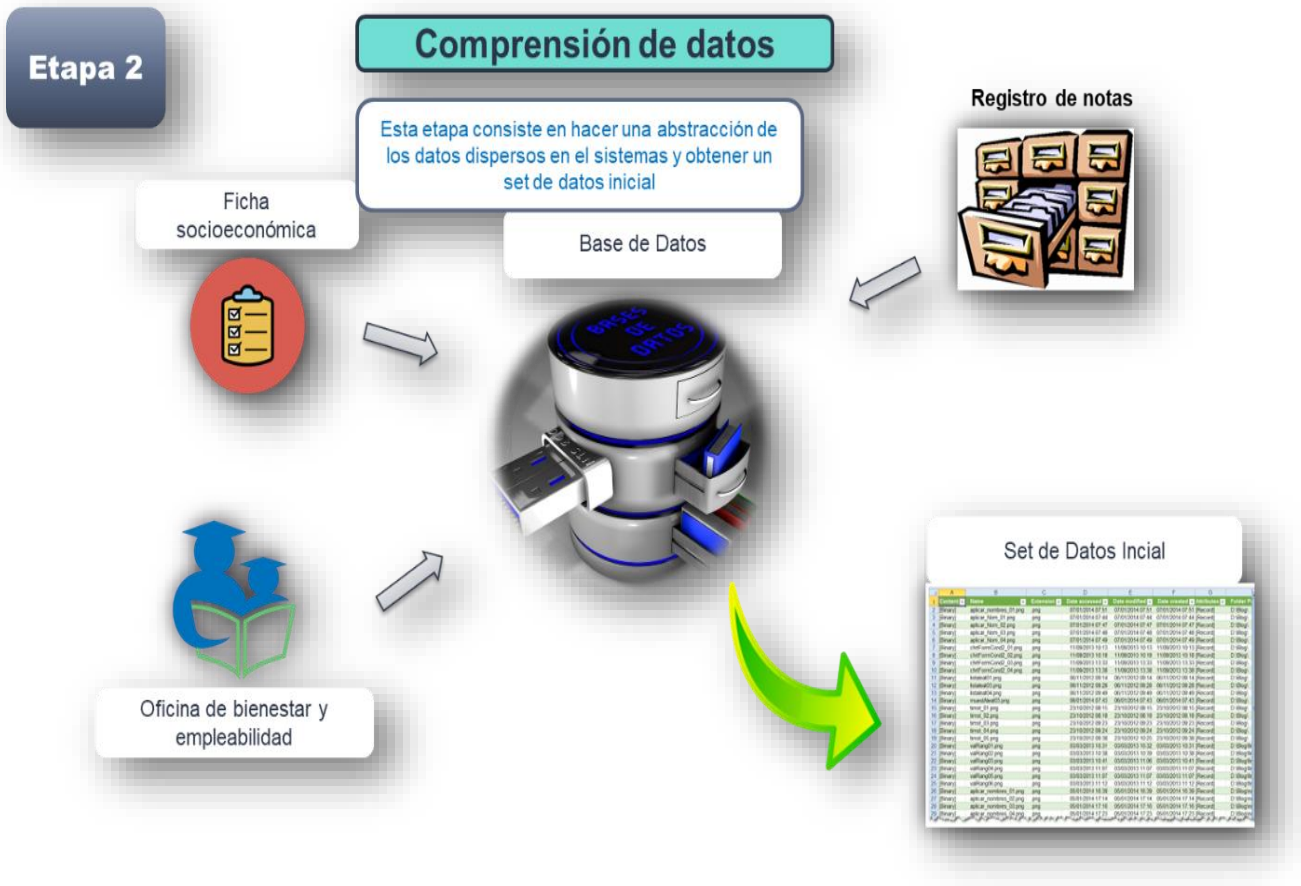
Tabla 8: Característica identificadas inicialmente para el dataset:

Nro	**CarFam**	**EdaEst**	**PromPon**	**Proced_Est (0=C;1=L)**	**Sit_Laboral (0=No;1=Si)**	**Ingreso_Fam**	**Vivienda (0=Propia;1=No)**	**Serv_Internet (0=No;1=Si)**
Indica orden secuencial	Indica la carga familiar del estudiante	Indica la edad del estudiante	Promedio ponderado del último periodo.	Procedencia. 0=Cerca del centro de estudios. 1=Fuera de distrito.	Situación Laboral. 0=No trabaja. 1=Si trabaja	Ingreso familiar en soles	Vivienda. 0= si es propia. 1=si es alquilada	Cuenta Servicio de Internet. 0=no, 1= si cuenta

Seguro_Salud (0=Sis;1=Essalud)	**Reg_Aliment (0=2Veces;1=3Veces)**	**Tiene_Discap (0=No;1=Si)**	**Con_Quien_Vive (0=Solo;1=Familia)**	**Desertor (0=No;1=Si)**
Tiene seguro de salud 0=Sis, 1=EsSalud	Régimen alimentario. 0=2 veces por día 1=3 veces pro día	¿Es discapacitado? 0= no, 1=Si cuenta con alguna discapacidad	Con quiens vive. 0=solo, 1=con familia.	Desertor. 0=No deserta, 1= si es desertor.

Figura 30

Comprensión de los datos. Otra perspectiva



4.1.3 Preparación de datos:

En primer lugar, se realizó primero la recopilación de datos que se encontraban en varios formatos tales fichas físicas y hojas de cálculo, estableciendo un único formato en Excel con 12 características de mayor impacto en la deserción como sostienen autores citados anteriormente.

Posteriormente se importó la data al cuaderno de Jupiter Notebook, donde se realizó el trabajo de la limpieza de datos, identificando columnas sin datos, valores nulos o algún dato perdido. Este proceso no fue muy complicado, porque las columnas en su mayoría, tenían datos completos y algunos pocos no correspondían con el tipo de dato de la columna, situación que se pudo solucionar en algunos casos asignando el promedio de los datos restantes para completar el dato erróneo. Depurada la data, otro aspecto a tomar en cuenta en la preparación de datos, fue el problema de la dimensionalidad, muy común

en proyectos de esta naturaleza; es decir dimensiones que muchas veces tienen poco o nada que ver con el problema; es decir poca significancia. En ese caso se tenía 12 dimensiones y había que establecer cuáles de ellas tienen mayor relación con la variable predictora y obtener un set de datos solo con las dimensiones que son relevantes para el modelo. Esto se logró con el análisis de correlación de Karl Pearson, el cual establece que los coeficientes de correlación son la expresión numérica que indica el grado de relación existente entre 2 variables.

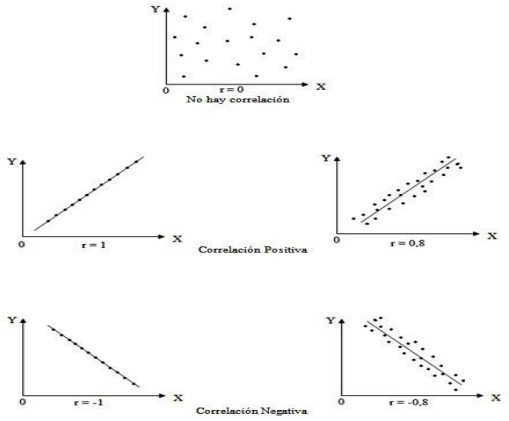
Sus valores varían entre los límites +1 y -1.

El valor r = 0 indica que no existe relación entre las variables.

El valor 1, indica una correlación perfecta positiva (al crecer o decrecer X, crece o decrece Y respectivamente). Un valor -1, indica una correlación perfecta negativa (Al crecer o decrecer X, decrece o crece Y respectivamente).

Figura 31

Correlación de Karl Pearson



Fuente: Nel, 2017.

Para interpretar el coeficiente de correlación utilizamos la siguiente escala:

Tabla 9: Tabla de correlación Pearson.

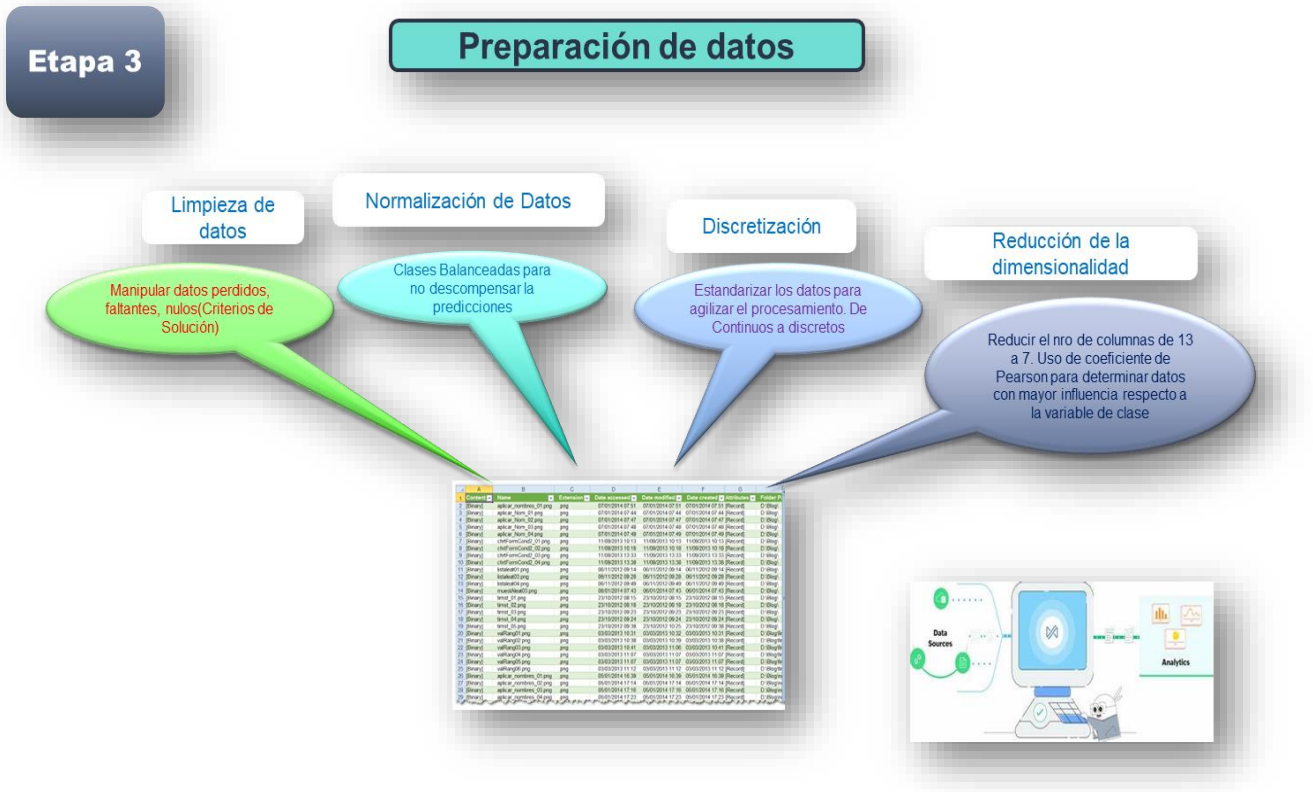
Valor	Significado
-1	*Correlación negativa grande y perfecta*
-0,9 a -0,99	*Correlación negativa muy alta*
-0,7 a -0,89	*Correlación negativa alta*
-0,4 a -0,69	*Correlación negativa moderada*
-0,2 a -0,39	*Correlación negativa baja*
-0,01 a -0,19	*Correlación negativa muy baja*
0	*Correlación nula*
0,01 a 0,19	*Correlación positiva muy baja*
0,2 a 0,39	*Correlación positiva baja*
0,4 a 0,69	*Correlación positiva moderada*
0,7 a 0,89	*Correlación positiva alta*
0,9 a 0,99	*Correlación positiva muy alta*
1	*Correlación positiva grande y perfecta*

En el presente trabajo se buscó dimensiones que tengan como mínimo una correlación moderada, estableciendo el índice de medición a 0.5, dado que este indicador va de 0.4 a 0.69.

A continuación, el procedimiento realizado.

Figura 32

Preparación de datos

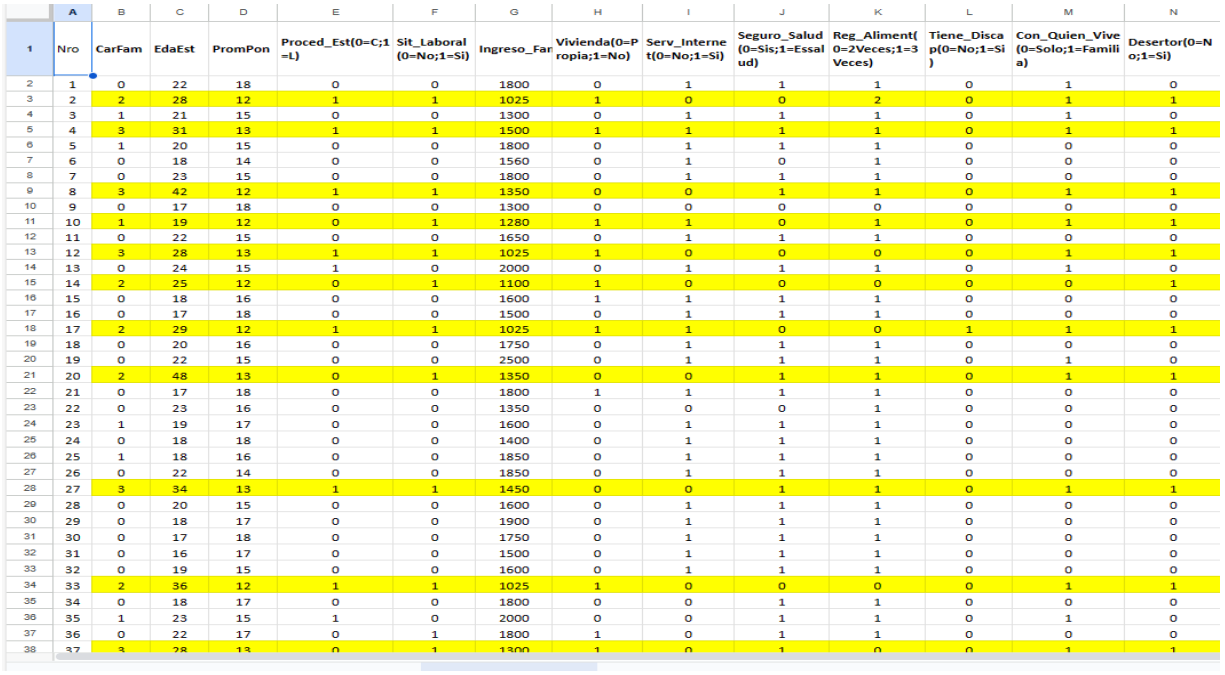


a.- Discretización de datos. – Estandarizar los datos de continuos a discretos. La variables categóricas se transforman a ceros(No) y Unos(Si), obteniendo el siguiente DataSet.

b.- Normalización de datos. – Se busca que las variables de clase estén balanceadas para garantizar la robustes del modelo. Para este caso verifico y transformo para algunos registros con el propósito que las clases no estén sesgadas para ningún extremo, esta situación podría llevar a producir un modelo que no refleje la realidad.

Figura 33

DataSet discretizada y normalizada



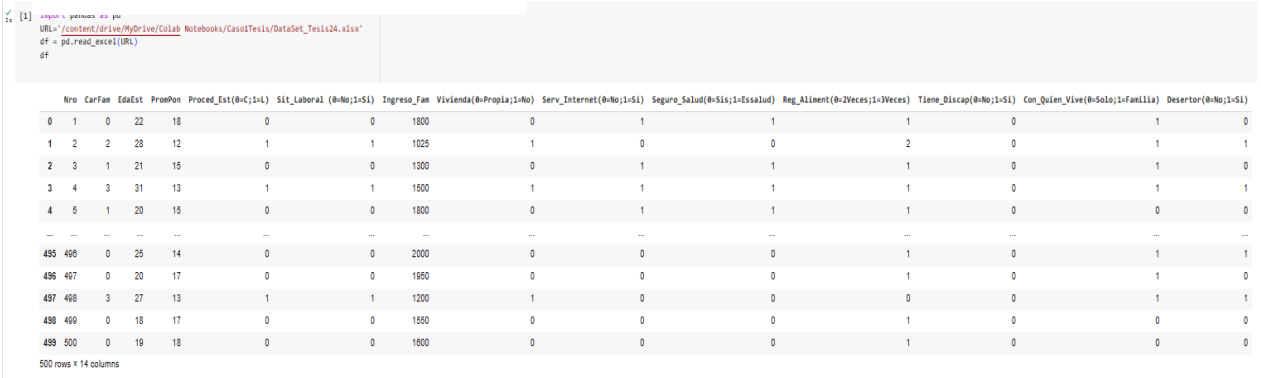
	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1	Nro	CarFam	EdaEst	PromPon	Proced_Est(0=C;1=L)	Sit_Laboral (0=No;1=Si)	Ingreso_Far	Vivienda(0=Propia;1=No)	Serv_Internet(0=No;1=Si)	Seguro_Salud (0=Sis;1=Essalud)	Reg_Aliment(0=2Veces;1=3Veces)	Tiene_Discap(0=No;1=Si)	Con_Quien_Vive (0=Solo;1=Familia)	Desertor(0=No;1=Si)
2	1	0	22	18	0	0	1800	0	1	1	1	0	1	0
3	2	2	28	12	1	1	1025	1	0	0	2	0	1	1
4	3	1	21	15	0	0	1300	0	1	1	1	0	1	0
5	4	3	31	13	1	1	1500	1	1	1	1	0	1	1
6	5	1	20	15	0	0	1800	0	1	1	1	0	0	0
7	6	0	18	14	0	0	1560	0	1	0	1	0	0	0
8	7	0	23	15	0	0	1800	0	1	1	1	0	0	0
9	8	3	42	12	1	1	1350	0	0	1	1	0	1	1
10	9	0	17	18	0	0	1300	0	0	0	0	0	0	0
11	10	1	19	12	0	1	1280	1	1	0	1	0	1	1
12	11	0	22	15	0	0	1650	0	1	1	1	0	0	0
13	12	3	28	13	1	1	1025	1	0	0	0	0	1	1
14	13	0	24	15	1	0	2000	0	1	1	1	0	1	0
15	14	2	25	12	0	1	1100	1	0	0	0	0	0	1
16	15	0	18	16	0	0	1600	1	1	1	1	0	0	0
17	16	0	17	18	0	0	1500	0	1	1	1	0	0	0
18	17	2	29	12	1	1	1025	1	1	0	0	1	1	1
19	18	0	20	16	0	0	1750	0	1	1	1	0	0	0
20	19	0	22	15	0	0	2500	0	1	1	1	0	1	0
21	20	2	48	13	0	1	1350	0	0	1	1	0	1	1
22	21	0	17	18	0	0	1800	1	1	1	1	0	0	0
23	22	0	23	16	0	0	1350	0	0	0	1	0	0	0
24	23	1	19	17	0	0	1600	0	1	1	1	0	0	0
25	24	0	18	18	0	0	1400	0	1	1	1	0	0	0
26	25	1	18	16	0	0	1850	0	1	1	1	0	0	0
27	26	0	22	14	0	0	1850	0	1	1	1	0	0	0
28	27	3	34	13	1	1	1450	0	0	1	1	0	1	1
29	28	0	20	15	0	0	1600	0	1	1	1	0	0	0
30	29	0	18	17	0	0	1900	0	1	1	1	0	0	0
31	30	0	17	18	0	0	1750	0	1	1	1	0	0	0
32	31	0	16	17	0	0	1500	0	1	1	1	0	0	0
33	32	0	19	15	0	0	1600	0	1	1	1	0	0	0
34	33	2	36	12	1	1	1025	1	0	0	0	0	1	1
35	34	0	18	17	0	0	1800	0	0	1	1	0	0	0
36	35	1	23	15	1	0	2000	0	0	1	1	0	1	0
37	36	0	22	17	0	1	1800	1	0	1	1	0	0	0
38	37	3	28	13	0	1	1300	1	0	1	0	0	1	1

c.- Limpieza de datos. – Manipular datos perdidos, faltantes, nulos (Criterios de Solución).

- **Importando los datos del archivo DataSet_Tesis24.xlsx**

Figura 34

Set de datos importado de una tabla



```
[1] import pandas as pd
URL='/content/drive/MyDrive/Colab Notebooks/Caso1Tesis/DataSet_Tesis24.xlsx'
df = pd.read_excel(URL)
df
```

	Nro	CarFam	EdaEst	PromPon	Proced_Est(0=C;1=L)	Sit_Laboral (0=No;1=Si)	Ingreso_Fam	Vivienda(0=Propia;1=No)	Serv_Internet(0=No;1=Si)	Seguro_Salud(0=Sis;1=Essalud)	Reg_Aliment(0=2Veces;1=3Veces)	Tiene_Discap(0=No;1=Si)	Con_Quien_Vive(0=Solo;1=Familia)	Desertor(0=No;1=Si)
0	1	0	22	18	0	0	1800	0	1	1	1	0	1	0
1	2	2	28	12	1	1	1025	1	0	0	2	0	1	1
2	3	1	21	15	0	0	1300	0	1	1	1	0	1	0
3	4	3	31	13	1	1	1500	1	1	1	1	0	1	1
4	5	1	20	15	0	0	1800	0	1	1	1	0	0	0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
495	496	0	25	14	0	0	2000	0	0	0	1	0	1	1
496	497	0	20	17	0	0	1950	0	0	0	1	0	1	0
497	498	3	27	13	1	1	1200	1	0	0	0	0	1	1
498	499	0	18	17	0	0	1550	0	0	0	1	0	0	0
499	500	0	19	18	0	0	1600	0	0	0	1	0	0	0

500 rows × 14 columns

- **Verificando que el data set no tenga algún dato nulo, vacíos o perdidos.**

Figura 35

Muestra datos completos

```
#Columnas sin datos
df.isna().sum()
```

```
Nro                                  0
CarFam                               0
EdaEst                               0
PromPon                              0
Proced_Est(0=C;1=L)                  0
Sit_Laboral (0=No;1=Si)              0
Ingreso_Fam                          0
Vivienda(0=Propia;1=No)              0
Serv_Internet(0=No;1=Si)             0
Seguro_Salud(0=Sis;1=Essalud)        0
Reg_Aliment(0=2Veces;1=3Veces)       0
Tiene_Discap(0=No;1=Si)              0
Con_Quien_Vive(0=Solo;1=Familia)     0
Desertor(0=No;1=Si)                  0
dtype: int64
```

- **Eliminando el campo Nro que no contribuye al modelo, debido a que es un valor de orden.**

Figura 36

Muestra eliminación del campo Nro

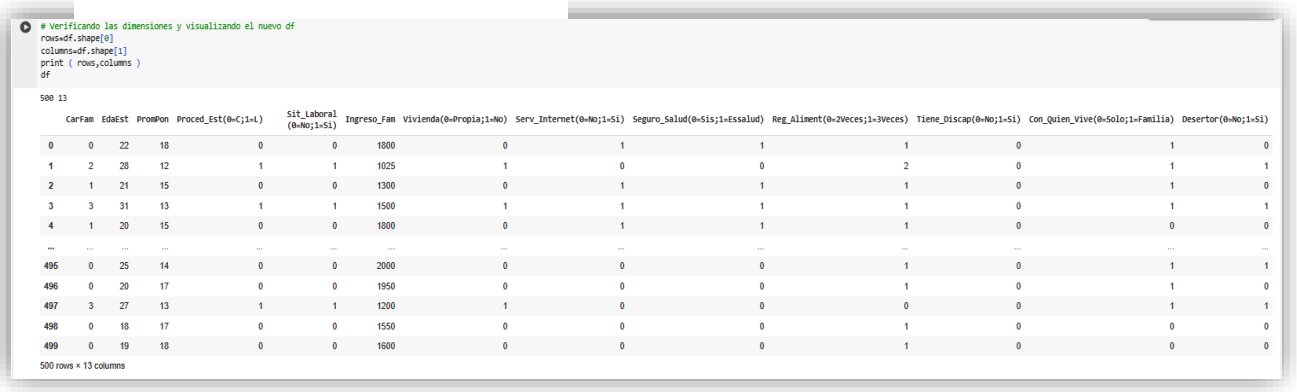
```
# Remover Columna "Nro" que no contribuye al modelo

df = df.drop('Nro', axis=1)
df.columns
```

```
Index(['CarFam', 'EdaEst', 'PromPon', 'Proced_Est(0=C;1=L)',
       'Sit_Laboral (0=No;1=Si)', 'Ingreso_Fam', 'Vivienda(0=Propia;1=No)',
       'Serv_Internet(0=No;1=Si)', 'Seguro_Salud(0=Sis;1=Essalud)',
       'Reg_Aliment(0=2Veces;1=3Veces)', 'Tiene_Discap(0=No;1=Si)',
       'Con_Quien_Vive(0=Solo;1=Familia)', 'Desertor(0=No;1=Si)'],
      dtype='object')
```

- **Verificando el nuevo DataSet sin el campo Nro.**

Figura 37

DataSet, sin campo Nro.



```
# Verificando las dimensiones y visualizando el nuevo df
rows=df.shape[0]
columns=df.shape[1]
print ( rows,columns )
df
```

```
500 13
```

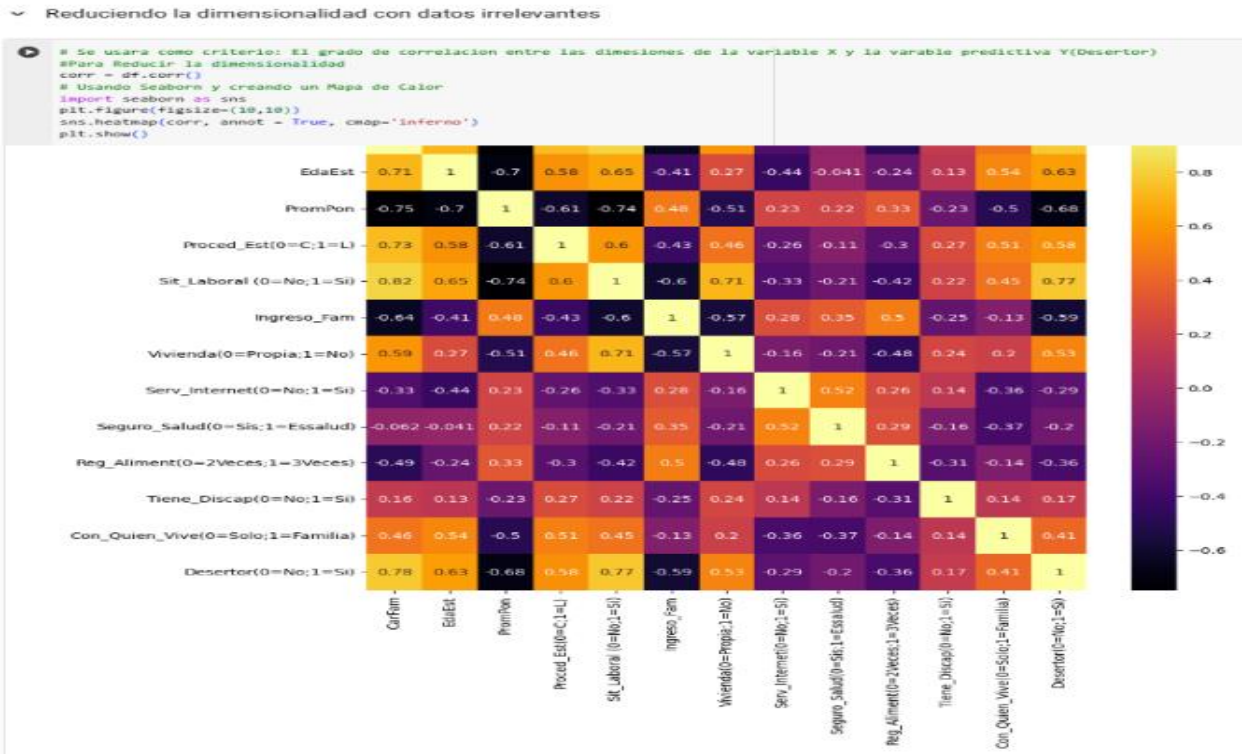
	CarFam	EdaEst	PromPon	Proced_Est(0=C;1=L)	Sit_Laboral (0=No;1=Si)	Ingreso_Fam	Vivienda(0=Propia;1=No)	Serv_Internet(0=No;1=Si)	Seguro_Salud(0=Sis;1=Essalud)	Reg_Aliment(0=2Veces;1=3Veces)	Tiene_Discap(0=No;1=Si)	Con_Quien_Vive(0=Solo;1=Familia)	Desertor(0=No;1=Si)
0	0	22	18	0	0	1800	0	1	1	1	0	1	0
1	2	28	12	1	1	1025	1	0	0	2	0	1	1
2	1	21	15	0	0	1300	0	1	1	1	0	1	0
3	3	31	13	1	1	1500	1	1	1	1	0	1	1
4	1	20	15	0	0	1800	0	1	1	1	0	0	0
...	...	...	...	...	...	...	...	...	...	...	...	...	...
495	0	25	14	0	0	2000	0	0	0	1	0	1	1
496	0	20	17	0	0	1950	0	0	0	1	0	1	0
497	3	27	13	1	1	1200	1	0	0	0	0	1	1
498	0	18	17	0	0	1550	0	0	0	1	0	0	0
499	0	19	18	0	0	1600	0	0	0	1	0	0	0

500 rows × 13 columns

d.- Reduciendo la dimensionalidad. – Reducir el número de columnas de 13 a 7. Uso de coeficiente de Pearson para determinar datos con mayor influencia respecto a la variable de clase.

Figura 38

Gráfico de Calor con las dimensiones originales



Reduciendo la dimensionalidad con datos irrelevantes

```
# Se usara como criterio: El grado de correlacion entre las dimesiones de la variable X y la varable predictiva Y(Desertor)
#Para Reducir la dimensionalidad
corr = df.corr()
# Usando Seaborn y creando un Mapa de Calor
import seaborn as sns
plt.figure(figsize=(10,10))
sns.heatmap(corr, annot = True, cmap='inferno')
plt.show()
```

Figura 39

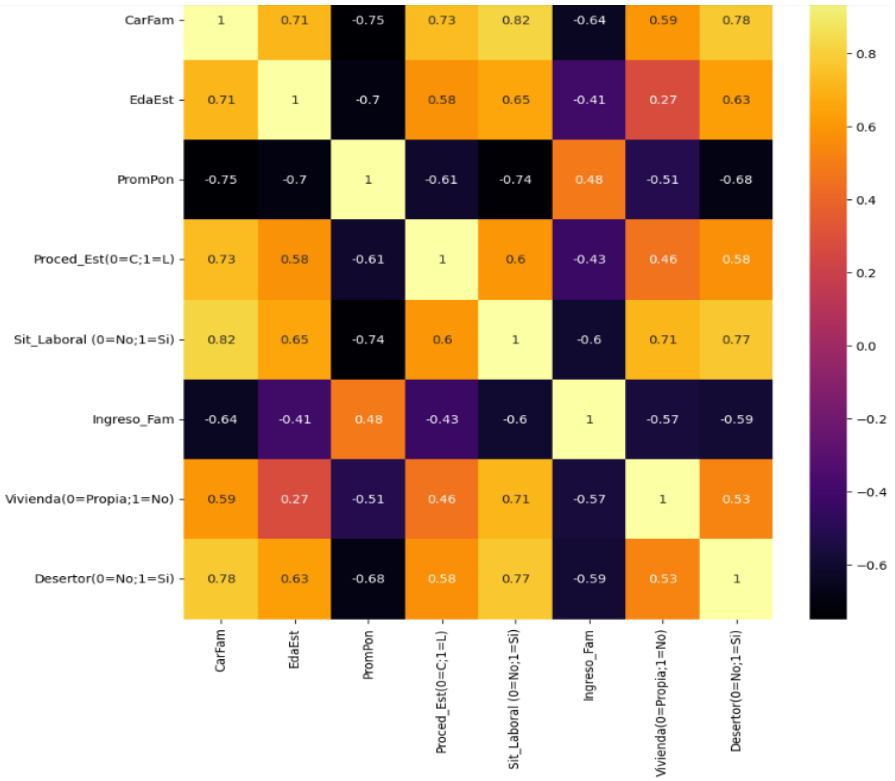
Gráfico indicando las variables que tiene correlación >0.5, reduciendo las dimensiones de 13 a 8, incluida la variable de clase.

```
[ ] # Seleccionar las columnas con mayor correlacion  respecto a la Desercion.
    corr[abs(corr['Desertor(0=No;1=Si)']) > 0.5]
```

	CarFam	EdaEst	PromPon	Proced_Est(0=C;1=L)	Sit_Laboral (0=No;1=Si)	Ingreso_Fam	Vivienda(0=Propia;1=No)	Desertor(0=No;1=Si)
CarFam	1.000000	0.712805	-0.749520	0.733252	0.821657	-0.642793	0.590623	0.780689
EdaEst	0.712805	1.000000	-0.700666	0.578953	0.653693	-0.405781	0.270281	0.627928
PromPon	-0.749520	-0.700666	1.000000	-0.607012	-0.740259	0.481199	-0.512677	-0.683085
Proced_Est(0=C;1=L)	0.733252	0.578953	-0.607012	1.000000	0.600533	-0.431427	0.455690	0.575684
Sit_Laboral (0=No;1=Si)	0.821657	0.653693	-0.740259	0.600533	1.000000	-0.599748	0.706441	0.772507
Ingreso_Fam	-0.642793	-0.405781	0.481199	-0.431427	-0.599748	1.000000	-0.565717	-0.586454
Vivienda(0=Propia;1=No)	0.590623	0.270281	-0.512677	0.455690	0.706441	-0.565717	1.000000	0.527800
Desertor(0=No;1=Si)	0.780689	0.627928	-0.683085	0.575684	0.772507	-0.586454	0.527800	1.000000

Figura 40

Gráfico de Calor con las dimensiones reducidas de 13 a 8 incluida la variable de clase



Como se puede apreciar en el gráfico, en la columna izquierda tenemos 7 dimensiones de la variable X y una para la variable Y. Aquellas dimensiones son las que tienen correlación respecto a la variable Y mayor a 0.5, ligeramente moderada según la tabla de Pearson.
Por lo tanto, la dimensionalidad se redujo de 12 a 7 en nuestro dataset.

Figura 41

DatSet definitivo

```
# El Set de Datos con el que se trabajara
df1
```

	CarFam	EdaEst	PromPon	Proced_Est(0=C;1=L)	Sit_Laboral (0=No;1=Si)	Ingreso_Fam	Vivienda(0=Propia;1=No)	Desertor(0=No;1=Si)
0	0	22	18	0	0	1800	0	0
1	2	28	12	1	1	1025	1	1
2	1	21	15	0	0	1300	0	0
3	3	31	13	1	1	1500	1	1
4	1	20	15	0	0	1800	0	0
...	...	...	...	...	...	...	...	...
495	0	25	14	0	0	2000	0	1
496	0	20	17	0	0	1950	0	0
497	3	27	13	1	1	1200	1	1
498	0	18	17	0	0	1550	0	0
499	0	19	18	0	0	1600	0	0

500 rows × 8 columns

4.1.4 Modelado:

En relación a la problemática del caso de estudio, en esta etapa se realizó la implementación de los modelos basados en Regresión Logística Binaria, Naive Bayes, Bosques Aleatorios y Máquinas de Soporte Vectorial en algoritmos de aprendizaje supervisadas usando python, pandas, numpy y scikit-learn.

Para este caso se entrenaron los algoritmos con el set de datos definitivo, para obtener los modelos deseados

Se evalúan la efectividad de los modelos obtenidos, los mismos que serán las entradas del modelo ensamblado.

Se ensambla luego un nuevo modelo a partir de los cuatro anteriores para mejorar la predicción, usando un algoritmo de ensamble voting clasifier, para máquinas de aprendizaje supervisados con el fin de verificar si mejora los resultados respecto al resto de algoritmos.

A través de la librería scikit-learn, procedemos a importar los algoritmos de clasificaciónque usaremos para instancias los modelos, implementarlos y evaluarlos.

a.- Se importa los algoritmos de aprendizaje, métricas y libreras graficas para la implementación de los modelos.

Figura 42

Importación de los algoritmos de aprendizaje, métricas y libreras gráficas para la implementación de los modelos

```
## importing models
from sklearn.ensemble import RandomForestClassifier
from sklearn.linear_model import LogisticRegression
from sklearn.naive_bayes import GaussianNB
from sklearn.svm import SVC
from sklearn.ensemble import VotingClassifier
from sklearn.metrics import accuracy_score
from sklearn.model_selection import train_test_split
import matplotlib.pyplot as plt
```

b.- Se verifica el DataSet que participará en el entrenamiento y prueba

Figura 43

DataSet seleccionado

```
# El Set de Datos con el que se trabajara
m
```

	CarFam	EdaEst	PromPon	Proced_Est(0=C;1=L)	Sit_Laboral (0=No;1=Si)	Ingreso_Fam	Vivienda(0=Propia;1=No)	Desertor(0=No;1=Si)
0	0	22	18	0	0	1800	0	0
1	2	28	12	1	1	1025	1	1
2	1	21	15	0	0	1300	0	0
3	3	31	13	1	1	1500	1	1
4	1	20	15	0	0	1800	0	0
...	...	...	...	...	...	...	...	...
495	0	25	14	0	0	2000	0	1
496	0	20	17	0	0	1950	0	0
497	3	27	13	1	1	1200	1	1
498	0	18	17	0	0	1550	0	0
499	0	19	18	0	0	1600	0	0

500 rows × 8 columns

c.- Se separa el data set en las variables X y la variable Y.

Figura 44

Separación del Set de Datos

```
X = df[caracteristicas]

y = df["Desertor(0=No;1=Si)"]

X
```

	CarFam	EdaEst	PromPon	Proced_Est(0=C;1=L)	Sit_Laboral (0=No;1=Si)	Ingreso_Fam	Vivienda(0=Propia;1=No)
0	0	22	18	0	0	1800	0
1	2	28	12	1	1	1025	1
2	1	21	15	0	0	1300	0
3	3	31	13	1	1	1500	1
4	1	20	15	0	0	1800	0
...	...	...	...	...	...	...	...
495	0	25	14	0	0	2000	0
496	0	20	17	0	0	1950	0
497	3	27	13	1	1	1200	1
498	0	18	17	0	0	1550	0
499	0	19	18	0	0	1600	0

500 rows × 7 columns

```
#Visualizando y
y
```

```
0      0
1      1
2      0
3      1
4      0
      ..
495    1
496    0
497    1
498    0
499    0
Name: Desertor(0=No;1=Si), Length: 500, dtype: int64
```

d.- Luego se procede a separar nuestra data en entrenamiento (80%) y prueba(20%)

Figura 45

Separación del Set de Datos en entrenamiento y prueba

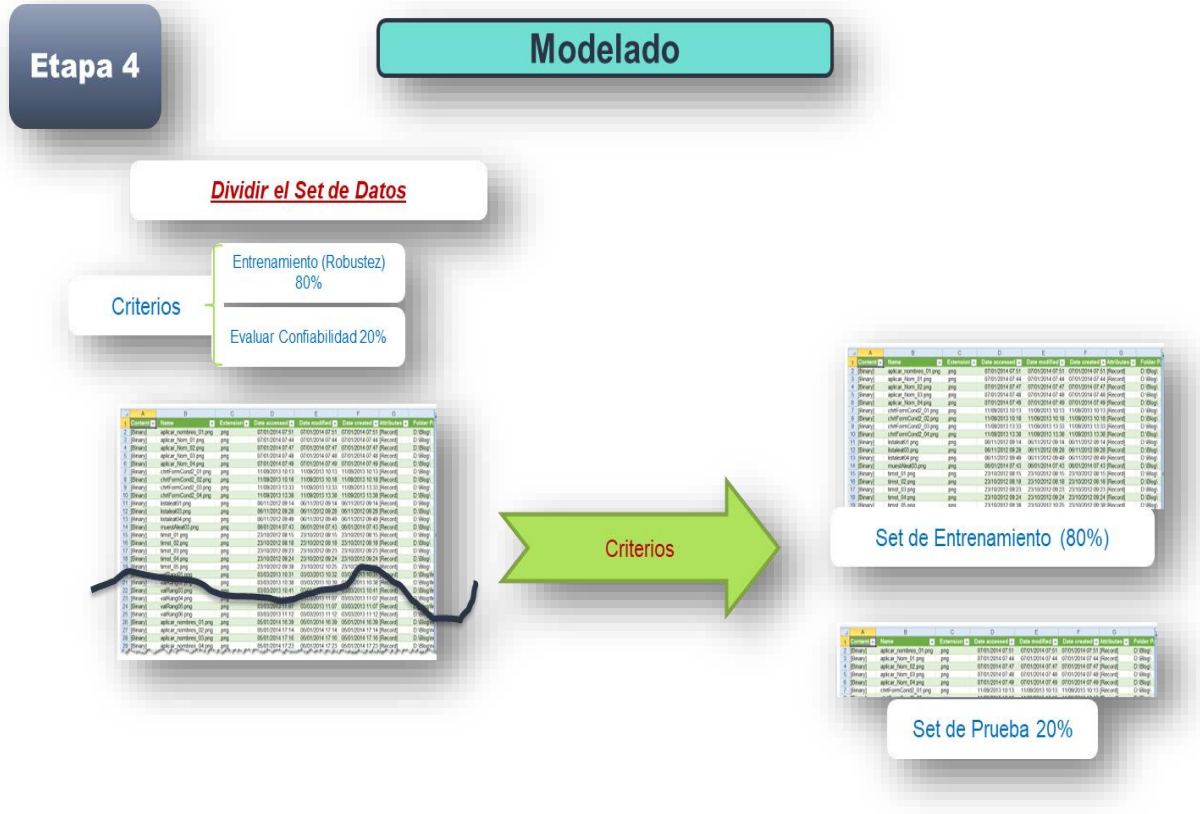


Figura 46

Separación del Set de Datos en entrenamiento y prueba (filas)

```
## Separando nuestra data en entrenamiento y prueba

X_train,X_test,y_train, y_test = train_test_split( x,y,test_size=.20,random_state=42323232)
```

e.- Se genera instancias para cada modelo

Modelo1(Regresión Logística Binaria).

Modelo2(Naive Bayes).

Modelo3(Bosques Aletorios).

Modelo 4(Maquinas de Soporte Vectorial).

f.- Se entrenan los algoritmos con el DataSet de entrenamiento para obtener los modelos predictivos

Figura 47

Entrenando algoritmo para obtener el modelo1

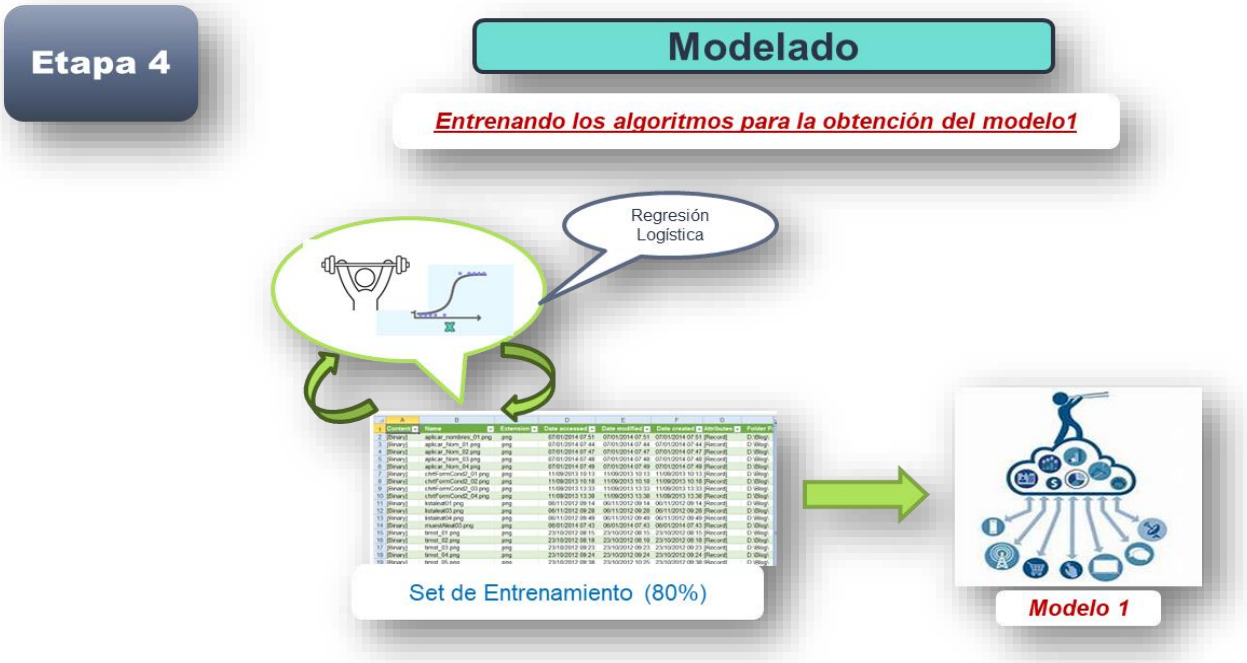


Figura 48

Entrenando algoritmo para obtener el modelo2

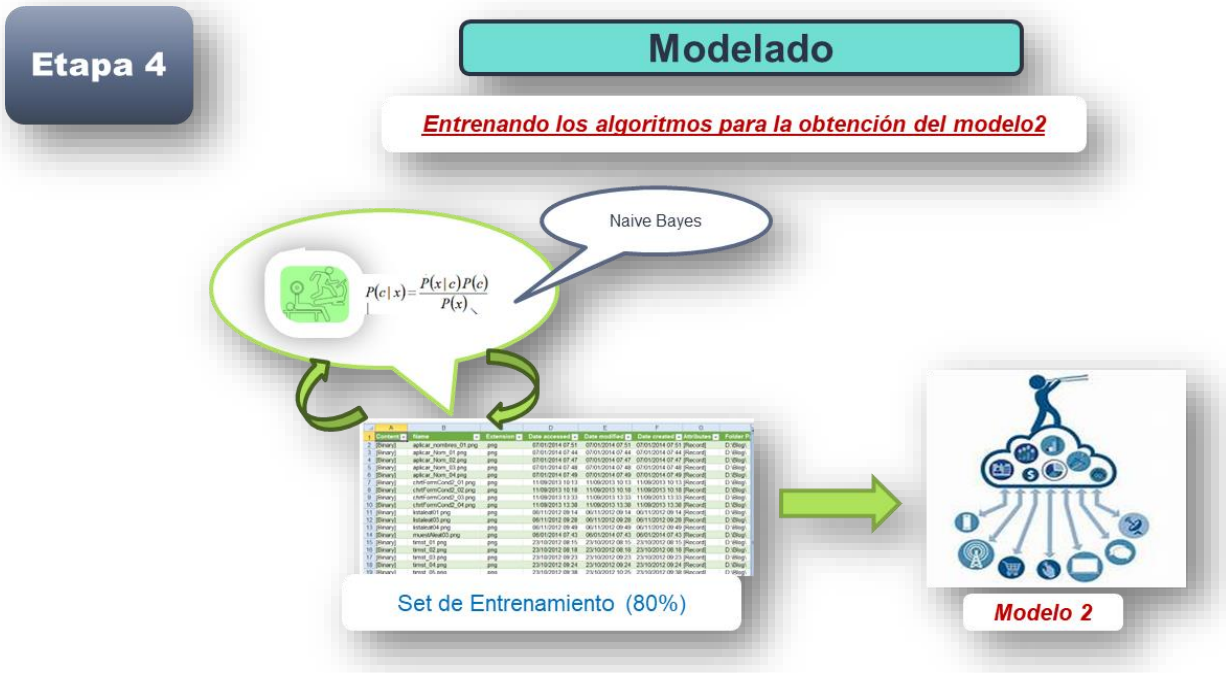


Figura 49

Entrenando algoritmo para obtener el modelo3

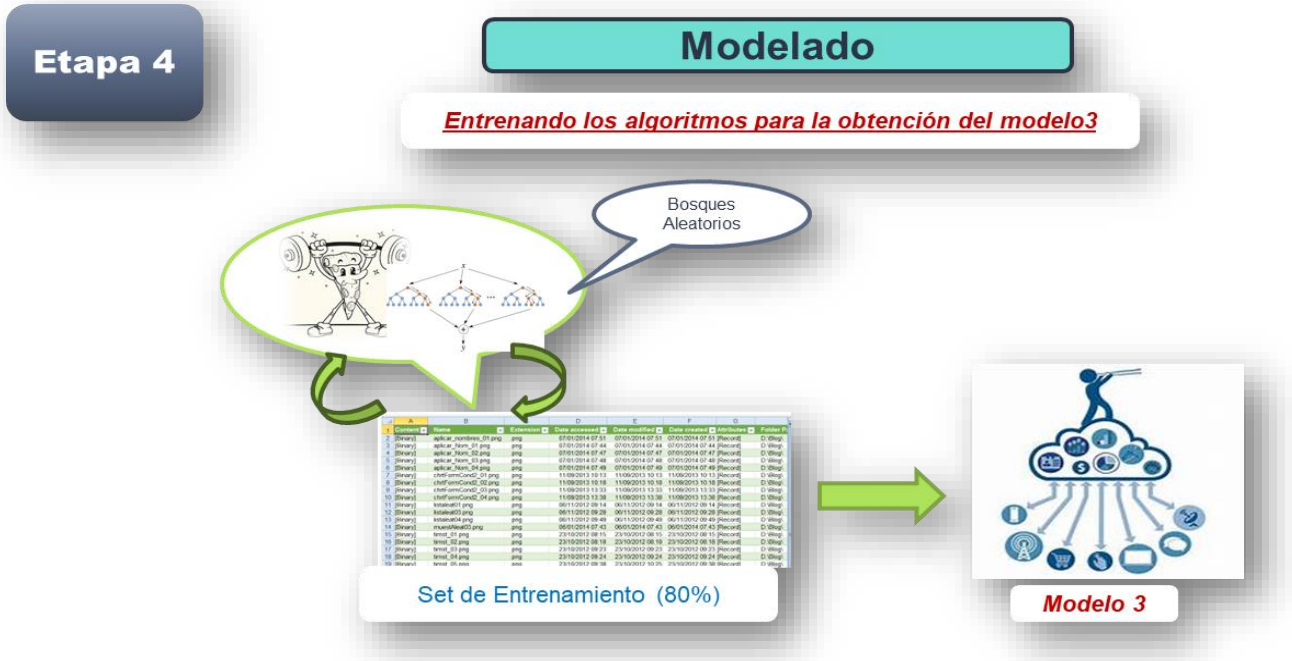
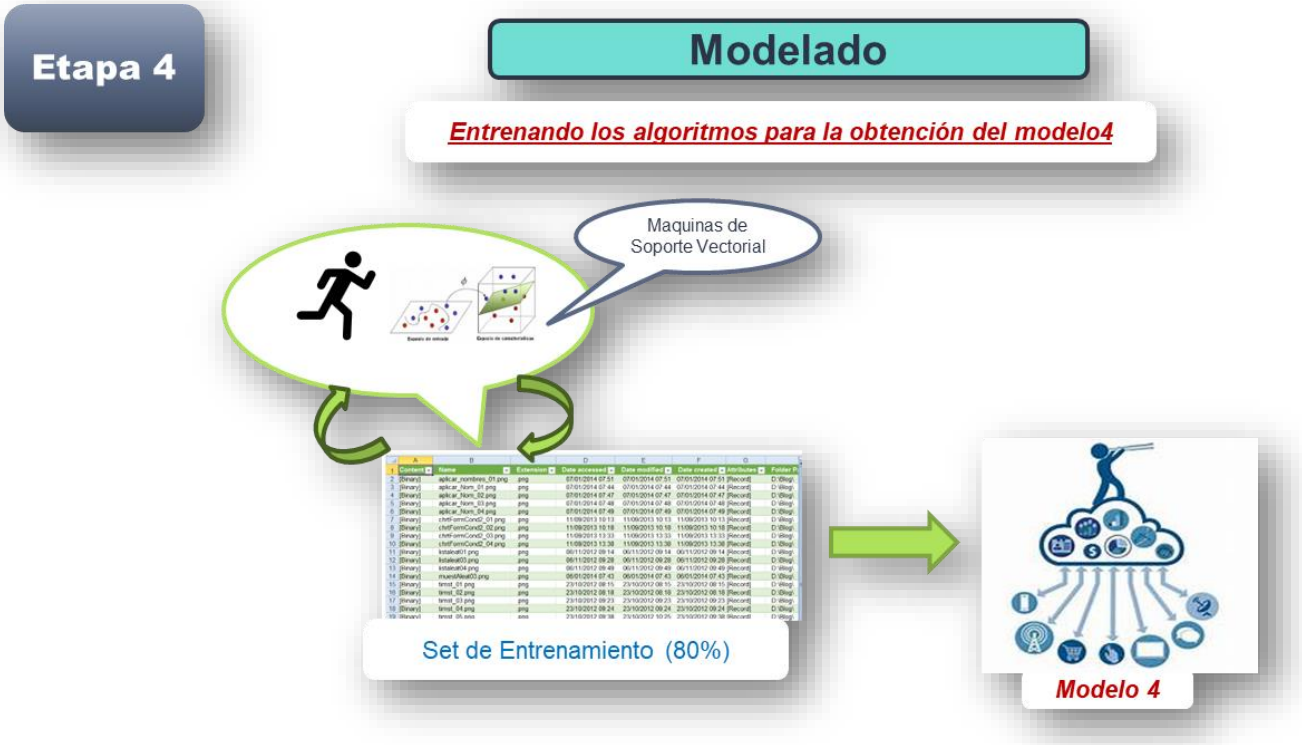


Figura 50

Entrenando algoritmo para obtener el modelo4



Con la idea de mejorar las confiabilidades anteriores, en la siguiente etapa se ensambla el modelo propuesto y evalúa el grado de confiabilidad. Se usará el algoritmos de ensamble Voting Classifier (Clasificador de votaciones).

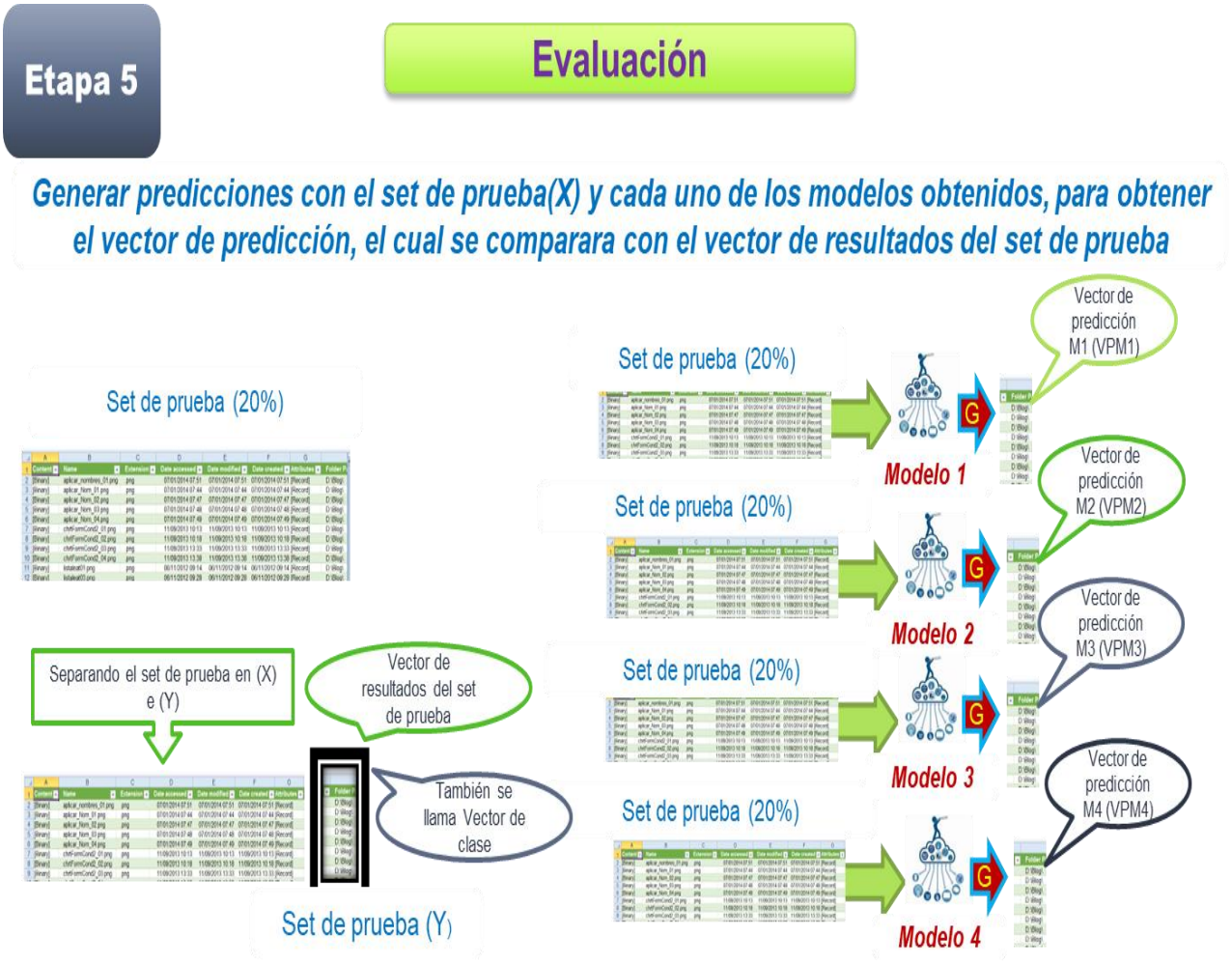
4.1.5 Evaluación:

En esta etapa se evalúa la confiabilidad de cada modelo, para ser usados como entradas en el modelo ensamblado. En esta etapa también detallamos la cantidad de estudiantes desertores y no desertores que el modelo reconoció correctamente y en cuantos se equivocó a través de la métrica denominada Matriz de Confusión.

a.- Generación de predicciones con cada modelo a partir del set de datos de prueba.

Figura 51

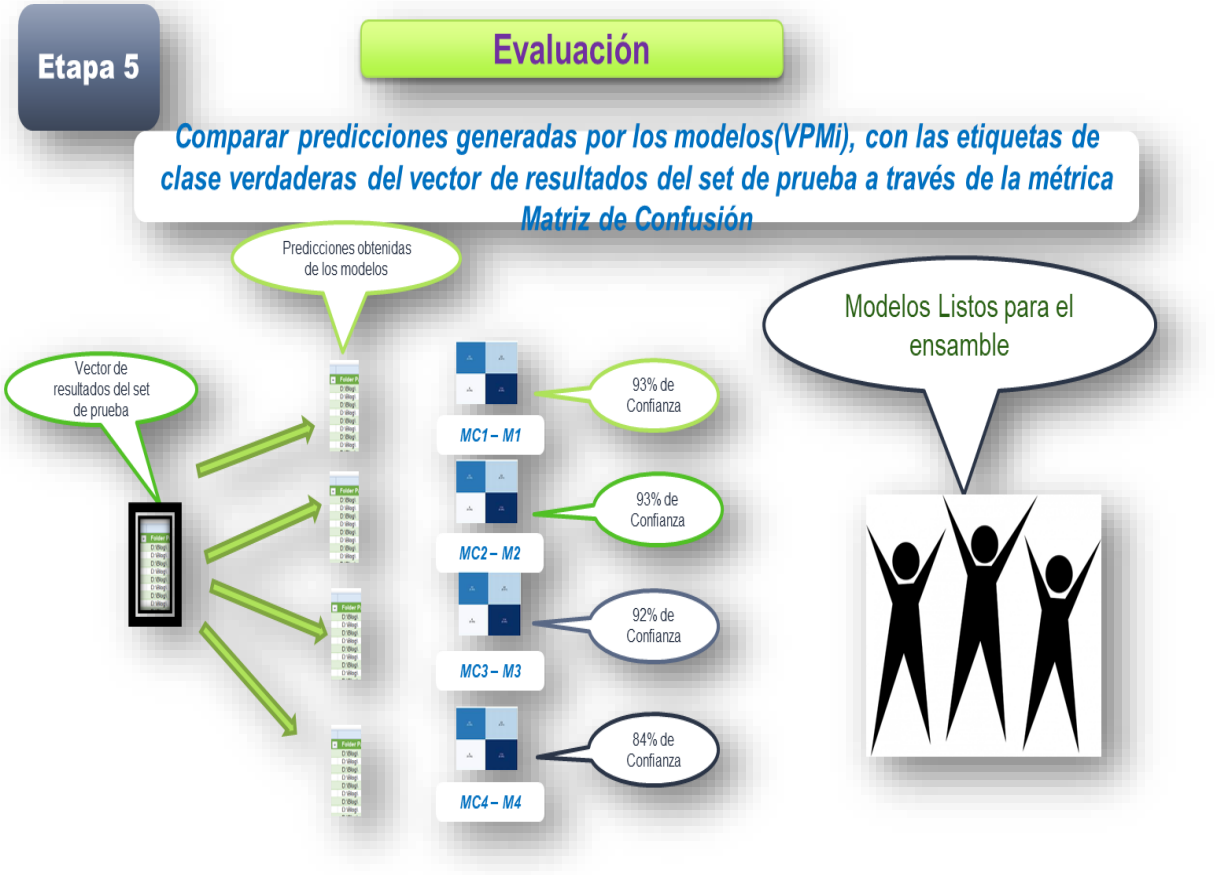
Muestra los vectores de predicción generados a partir del set de prueba



b.- Evaluar confiabilidad de los modelos creados a través de la matriz de confusión.

Figura 52

Muestra los porcentajes de confiabilidad de cada modelo



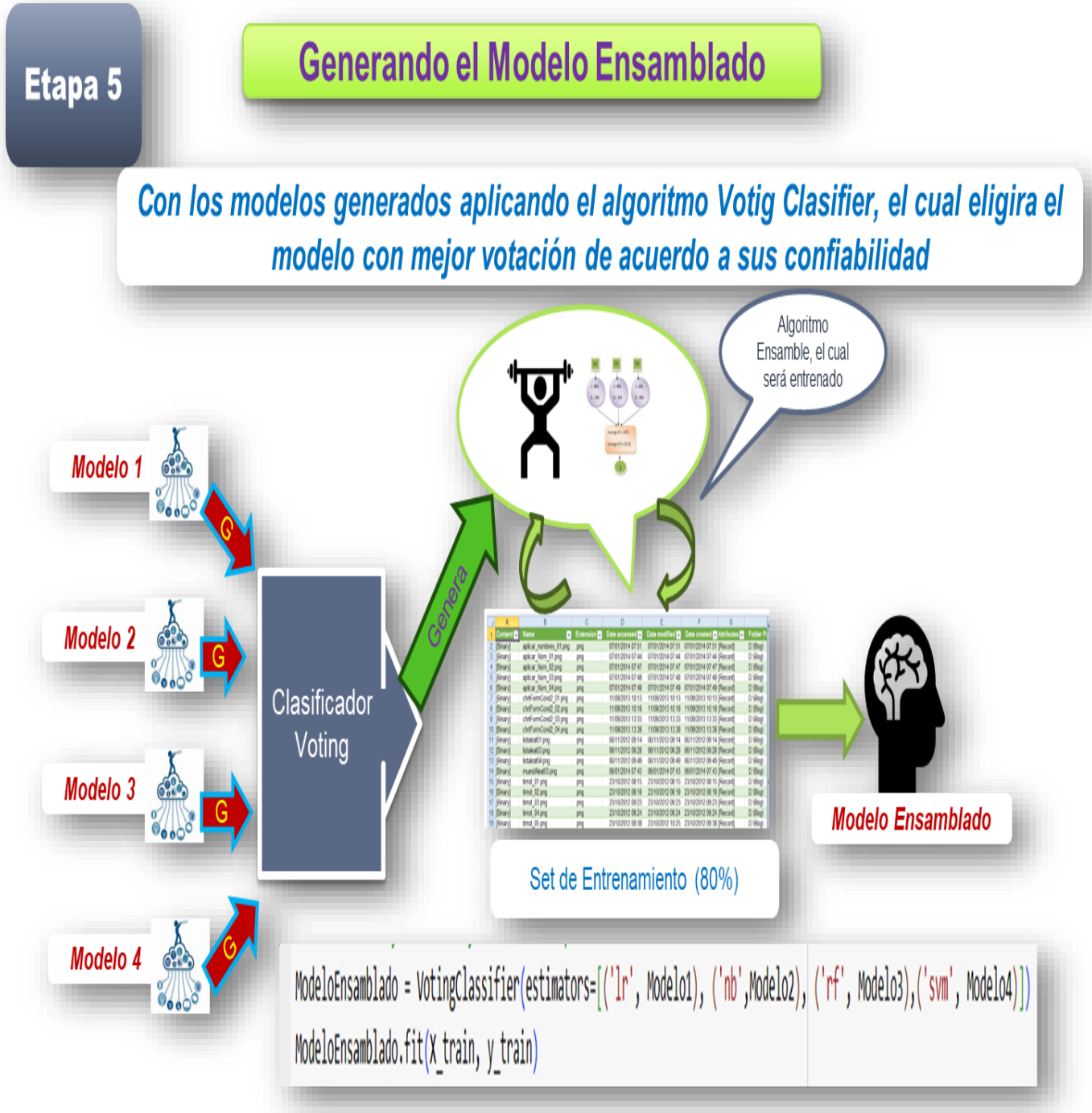
Con la idea de mejorar las confiabilidades anteriores, se procede a ensamblar un modelo para ver si podría aun mejorarse el grado de confiabilidad, usando para ello algoritmos de ensamble Voting Classifier (Clasificador de votaciones).

Figura 53
Modelo Ensamblado con Voting Classifier

```
[27]
    voting_clf = VotingClassifier(estimators=[('lr', modelo1),('rf',modelo2), ('svm', modelo3), ('nb', modelo4)])

    for clf in (modelo1, modelo2, modelo3, modelo4, voting_clf):
        clf.fit(X_train, y_train)
        y_pred = clf.predict(X_test)
        print(clf.__class__.__name__, accuracy_score(y_test, y_pred)*100 ,"%")

    LogisticRegression 93.0 %
    RandomForestClassifier 92.0 %
    SVC 84.0 %
    GaussianNB 93.0 %
    VotingClassifier 93.0 %
```

c.- Generar el Modelo ensamblado.

Figura 54
Obteniendo el modelo ensamblado



d.- Probar el modelo Ensamblado.

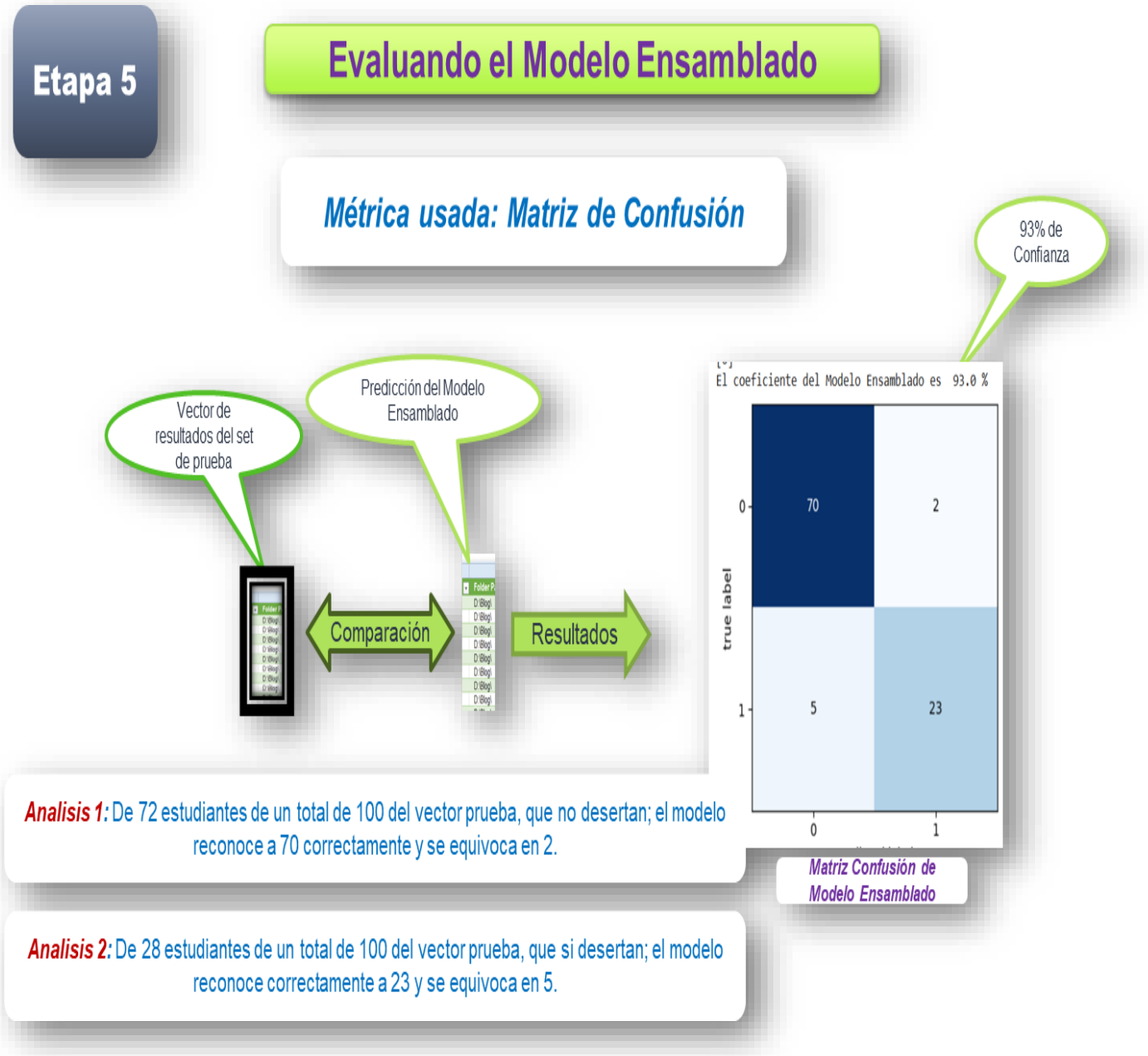
Figura 55

Haciendo una predicción con el modelo ensamblado.

e.- Evaluación del Modelo Ensamblado.

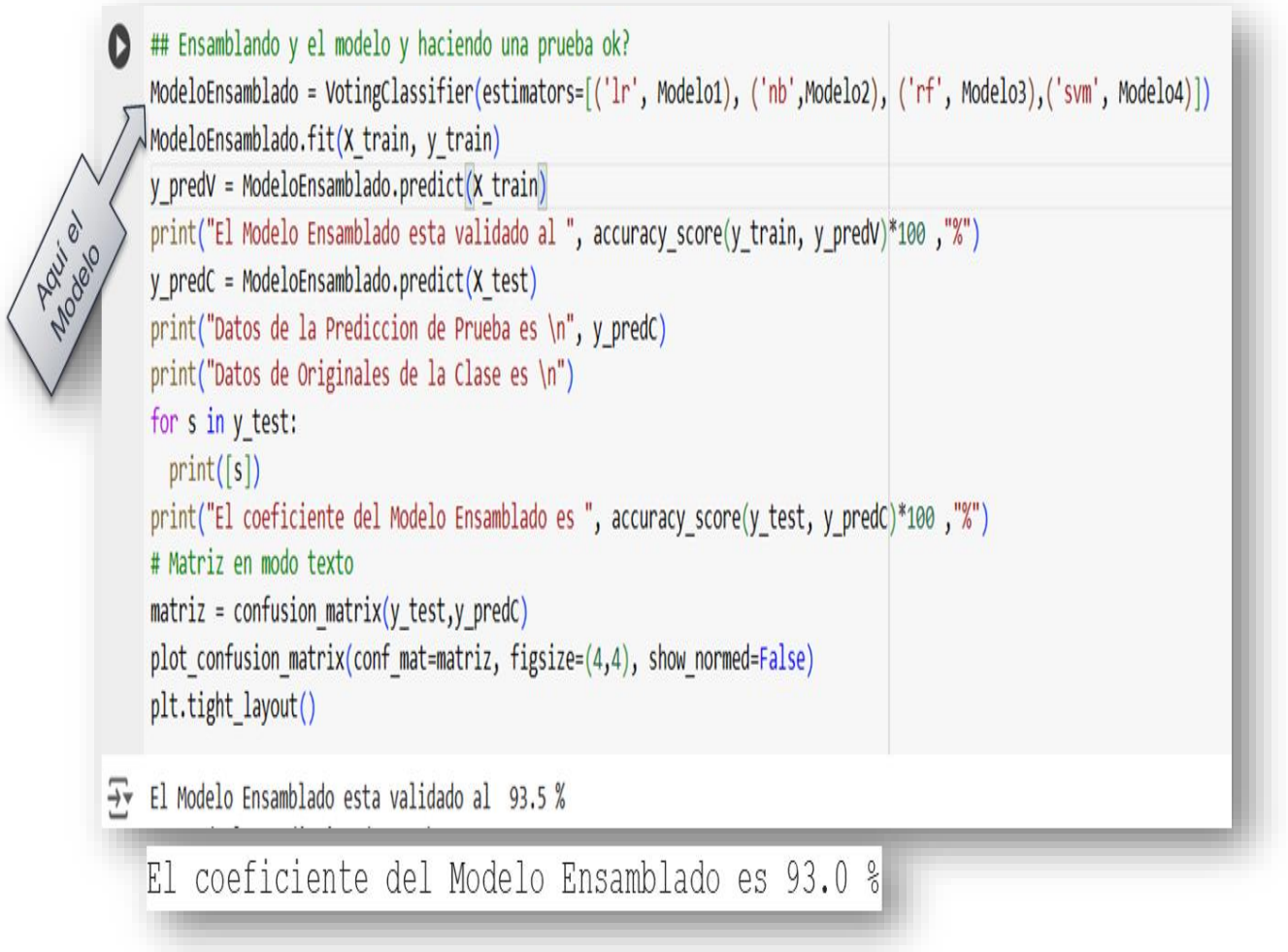
Figura 56

Predicción con el modelo ensamblado



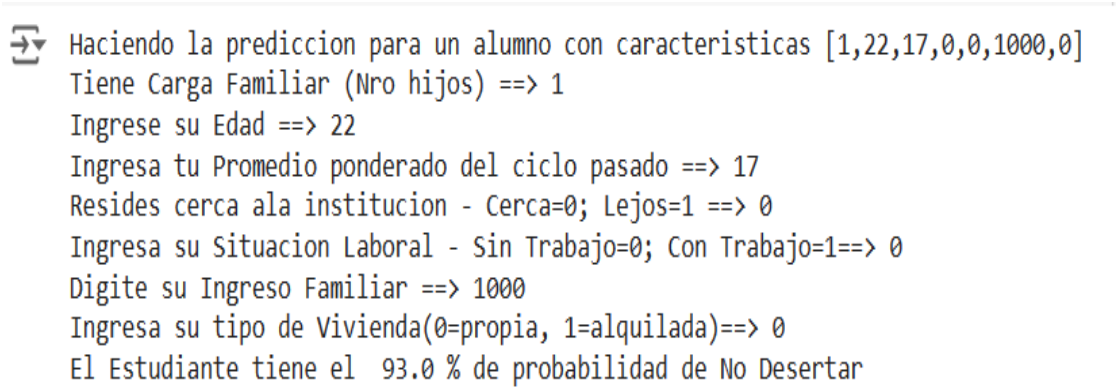
f.- Codificación del Modelo Ensamblado.

Figura 57
Codificando el modelo ensamblado



g.- Probado el modelo con datos de un estudiante.

Figura 58
Prueba de modelo con datos de un estudiante



Resumen de análisis de confiabilidad de los Modelos

a) Modelo 1, basado en regresión logística binaria

Figura 59*: Modelo 1, basado en Regresión logística Binaria.*

```
# Prueba1 - Modelo Regresion Logistica Binaria
from mlxtend.plotting import plot_confusion_matrix
from sklearn.metrics import confusion_matrix
ypred1=modelo1.predict(X_train)
ypred2=modelo1.predict(X_test)
print("Datos de la Prediccion",ypred2)
# Matriz en modo texto
matriz = confusion_matrix(y_test,ypred2)
plot_confusion_matrix(conf_mat=matriz, figsize=(4,4), show_normed=False)
plt.tight_layout()
print("El % de Confiabilidad del Modelo Entrenamiento de Regresion Logistica Binaria es ",accuracy_score(y_train, ypred1)*100 ,"%")
print("El % de Confiabilidad del Modelo Prueba de Regresion Logistica Binaria es ",accuracy_score(y_test, ypred2)*100 ,"%")
```

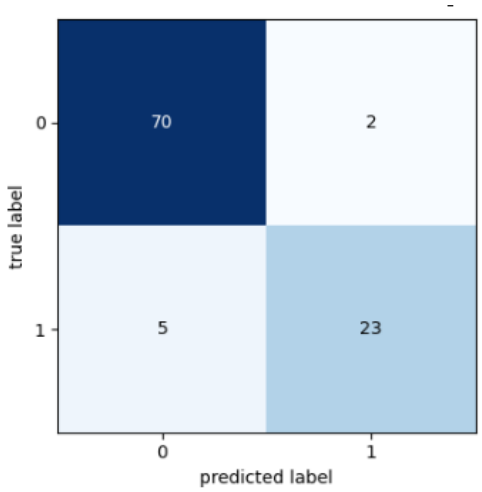
```
Datos de la Prediccion [0 1 0 0 0 1 1 0 0 0 0 1 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 1 0 0 1 0 0 1 0 0 0
 0 0 0 0 0 0 0 1 0 0 1 0 0 0 1 0 0 0 1 0 0 0 0 0 1 1 0 0 1 0 0 0 0 0 0 0 0
 0 1 0 1 0 0 0 0 0 0 1 1 0 1 0 0 1 1 1 1 1 0 0 0 0 0 0]
El % de Confiabilidad del Modelo Entrenamiento de Regresion Logistica Binaria es  93.5 %
El % de Confiabilidad del Modelo Prueba de Regresion Logistica Binaria es  93.0 %
```

Se puede ver que en el bloque de datos de entrenamiento nos ofrece un 93.5%, esto representa la validación o robustez del modelo y en el bloque de datos de prueba 93.0 % de confiabilidad.

En la métrica denominado matriz de confusión, analizaremos la cantidad de elementos que reconoce correctamente; es decir estudiantes que no desertan (0) y estudiantes que si desertan (1).

Figura 60

Matriz de confusión del modelo 1 regresión logística binaria



Análisis de la Matriz de confusión:

De 72 estudiantes que no desertan en la prueba de estudio, el modelo 1, reconoce a 70 y se equivoca en 2 estudiantes. Sostiene que 2 si desertan.

De 28 estudiantes que, si desertan en la prueba de estudio, el modelo 1, reconoce a 23 y no reconoce a 5. Sostiene que 5 no desertan.

Error en la predicción=7 estudiantes.

b) Modelo 2, basado en Naive Bayes

Figura 61
Modelo Naive Bayes

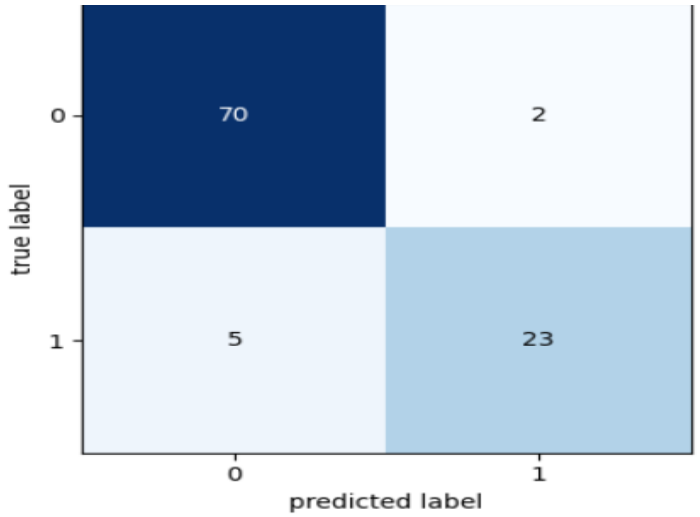
```
# Prueba4 Naive Bayes
from mlxtend.plotting import plot_confusion_matrix
from sklearn.metrics import confusion_matrix
ypred1=modelo4.predict(X_train)
ypred2=modelo4.predict(X_test)
print("Datos de la Prediccion",ypred2)
# Matriz en modo texto
matriz = confusion_matrix(y_test,ypred2)
plot_confusion_matrix(conf_mat=matriz, figsize=(4,4), show_normed=False)
plt.tight_layout()
print("El Porcentaje de Confiabilidad del Modelo Entrenamiento Naive Bayes es ",accuracy_score(y_train, ypred1)*100 ,"%")
print("El Porcentaje de Confiabilidad del Modelo Prueba Naive Bayes es ",accuracy_score(y_test, ypred2)*100 ,"%")
```

```
Datos de la Prediccion [0 1 0 0 0 1 1 0 0 0 0 1 0 0 0 1 1 0 0 0 0 0 0 0 0 0 1 0 0 1 0 0 1 0 0 0
 0 0 0 0 0 0 0 1 0 0 1 0 0 0 1 0 0 0 1 0 0 0 0 0 1 1 0 0 1 0 0 0 0 0 0 0 0
 0 1 0 1 0 0 0 0 0 0 1 1 0 1 0 0 1 1 1 1 0 0 0 0 0 0]
El Porcentaje de Confiabilidad del Modelo Entrenamiento Naive Bayes es  93.5 %
El Porcentaje de Confiabilidad del Modelo Prueba Naive Bayes es  93.0 %
```

Se puede ver que en el bloque de datos de entrenamiento nos ofrece un 93.5%, esto representa la validación o robustez del modelo y en el bloque de datos de prueba 93 % de confiabilidad.

En la métrica denominado matriz de confusión, analizaremos la cantidad de elementos que reconoce correctamente; es decir estudiantes que no desertan (0) y estudiantes que si desertor (1).

Figura 62 *Matriz de confusión del modelo 2*



Análisis de la Matriz de confusión:

De 72 estudiantes que no desertan en la prueba de estudio, el modelo 4, reconoce a 70 y se equivoca en 2 estudiantes. Sostiene que 2 si desertan.

De 28 estudiantes que si desertan en la prueba de estudio, el modelo 4, reconoce a 23 y no reconoce a 5. Sostiene que 5 no desertan.

Error en la predicción=7 estudiantes.

c) Modelo 3, basado en Bosque Aleatorios

Figura 63

Modelo 3 de Bosques Aleatorios

```
[ ] ### prueba2 Modelo Bosques aleatorios
    from mlxtend.plotting import plot_confusion_matrix
    from sklearn.metrics import confusion_matrix
    ypred1=modelo2.predict(X_train)
    ypred2=modelo2.predict(X_test)
    print("Datos de la Prediccion",ypred2)
    # Matriz en modo texto
    matriz = confusion_matrix(y_test,ypred2)
    plot_confusion_matrix(conf_mat=matriz, figsize=(4,4), show_normed=False)
    # show_normed=False
    plt.tight_layout()
    print("El Porcentaje de Confiabilidad del Modelo Entrenamiento de Bosques Aleatorios es ",accuracy_score(y_train, ypred1)*100 ,"%")
    print("El Porcentaje de Confiabilidad del Modelo Prueba de Bosques Aleatorios es ",accuracy_score(y_test, ypred2)*100 ,"%")
```

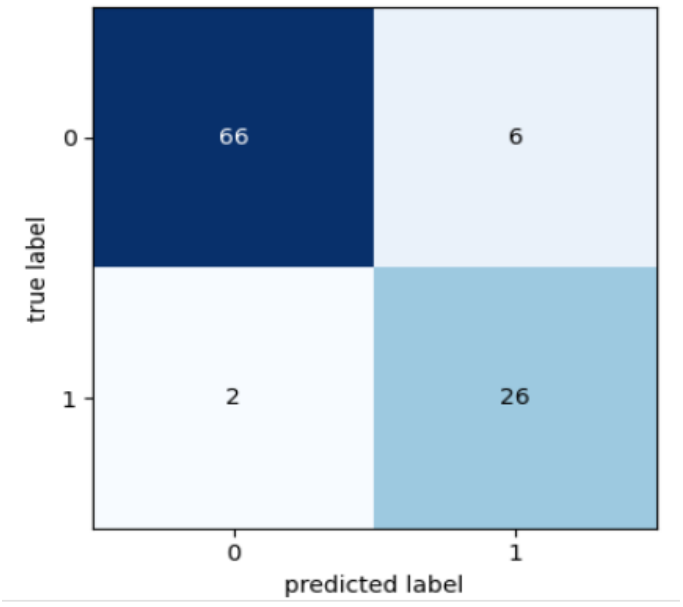
```
Datos de la Prediccion [0 1 0 0 1 1 1 0 0 0 0 1 1 0 0 1 1 1 0 0 1 0 0 1 0 0 0 1 0 0 1 0 0 1 0 0 0
 0 0 0 0 0 0 0 1 0 0 1 0 0 0 1 0 0 0 1 1 0 0 0 0 1 1 0 0 1 0 0 1 0 0 0 0 0 0
 0 1 0 1 0 0 0 0 0 0 1 1 0 1 0 0 1 1 1 1 1 0 0 0 0 0 0]
El Porcentaje de Confiabilidad del Modelo Entrenamiento de Bosques Aleatorios es  93.75 %
El Porcentaje de Confiabilidad del Modelo Prueba de Bosques Aleatorios es  92.0 %
```

Se puede ver que en el bloque de datos de entrenamiento nos ofrece un 93.75%, esto representa la validación o robustez del modelo y en el bloque de datos de prueba 92.0 % de confiabilidad.

En la métrica denominado matriz de confusión, analizaremos la cantidad de elementos que reconoce correctamente; es decir estudiantes que no desertan (0) y estudiantes que si desertan (1).

Figura 64

Matriz de confusión del modelo 2



Análisis de la Matriz de confusión:

De 72 estudiantes que no desertan en la prueba de estudio, el modelo 3, reconoce a 66 y se equivoca en 6 estudiantes. Sostiene que 6 si desertan.

De 28 estudiantes que si desertan en la prueba de estudio, el modelo 3, reconoce a 26 y no reconoce a 2. Sostiene que 2 no desertan.

Error en la predicción=8 estudiantes.

d) Modelo 4, basado en Maquinas de Soporte Vectorial.

Figura 65

Modelo 3 de Clasificador de Soporte Vectorial

```
from mlxtend.plotting import plot_confusion_matrix
from sklearn.metrics import confusion_matrix
ypred1=modelo3.predict(X_train)
ypred2=modelo3.predict(X_test)
print("Datos de la Prediccion",ypred2)
# Matriz en modo texto
matriz = confusion_matrix(y_test,ypred2)
plot_confusion_matrix(conf_mat=matriz, figsize=(4,4), show_normed=False)
# show_normed=False
plt.tight_layout()
print("El Porcentaje de Confiabilidad del Modelo Entrenamiento del Clasificador de Soporte Vectorial es ",accuracy_score(y_train, ypred1)*100 ,"%")
print("El Porcentaje de Confiabilidad del Modelo Prueba del Clasificador de Soporte Vectorial es ",accuracy_score(y_test, ypred2)*100 ,"%")
```

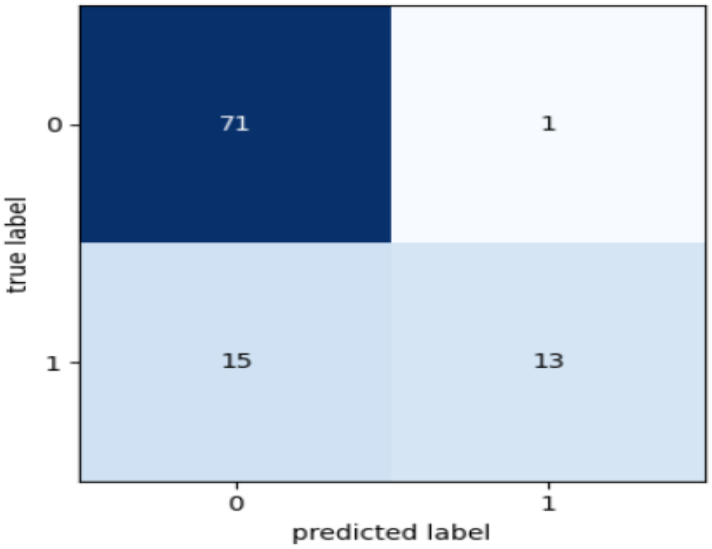
```
Datos de la Prediccion [0 1 0 0 0 0 1 0 0 0 0 1 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0
 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0
 0 0 0 1 0 0 0 0 0 0 0 1 0 0 1 0 0 1 0 1 0 0 0 0 0 0 0 0]
El Porcentaje de Confiabilidad del Modelo Entrenamiento del Clasificador de Soporte Vectorial es  83.75 %
El Porcentaje de Confiabilidad del Modelo Prueba del Clasificador de Soporte Vectorial es  84.0 %
```

Se puede ver que en el bloque de datos de entrenamiento nos ofrece un 83.75%, esto representa la validación o robustez del modelo y en el bloque de datos de prueba 84.0 % de confiabilidad.

En la métrica denominado matriz de confusión, analizaremos la cantidad de elementos que reconoce correctamente; es decir estudiantes que no desertan (0) y estudiantes que si desertan (1).

Figura 66

Matriz de confusión del modelo 4



Análisis de la Matriz de confusión:

De 72 estudiantes que no desertan en la prueba de estudio, el modelo 4, reconoce a 71 y se equivoca en 1 estudiante. Sostiene que 1 si deserta.

De 28 estudiantes que si desertan en la prueba de estudio, el modelo 4, reconoce a 13 y no reconoce a 15. Sostiene que 15 no desertan.

Error en la predicción=16 estudiantes.

e) Modelo Ensamblado.

Figura 67

Modelo Ensamblado con voting

```
## Es una Prueba de Ensamblado ok?
ModeloEnsamblado = VotingClassifier(estimators=[('lr', Modelo1), ('nb',Modelo2), ('rf', Modelo3),('svm', Modelo4)],  voting='hard')
ModeloEnsamblado.fit(X_train, y_train)
y_predV = ModeloEnsamblado.predict(X_train)
print("El Modelo Ensamblado esta validado al ", accuracy_score(y_train, y_predV)*100 ,"%")
y_predC = ModeloEnsamblado.predict(X_test)
print("Datos de la Prediccion de Prueba es \n", y_predC)
print("El coeficiente del Modelo Ensamblado es ", accuracy_score(y_test, y_predC)*100 ,"%")
# Matriz en modo texto
matriz = confusion_matrix(y_test,y_predC)
plot_confusion_matrix(conf_mat=matriz, figsize=(4,4), show_normed=False)
plt.tight_layout()
```

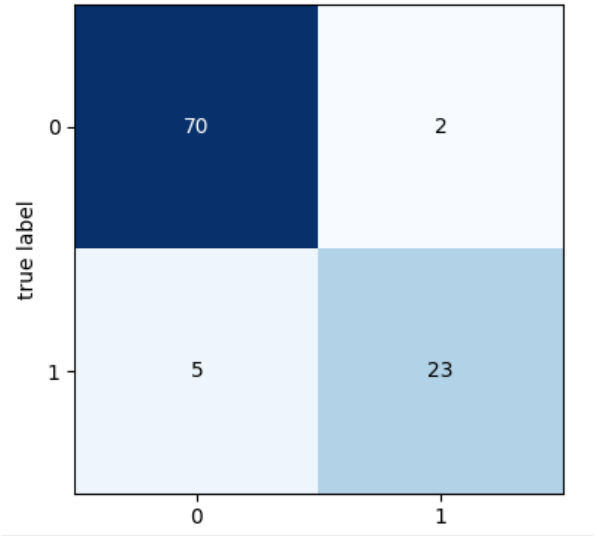
```
El Modelo Ensamblado esta validado al  93.5 %
Datos de la Prediccion de Prueba es
 [0 1 0 0 0 1 1 0 0 0 0 1 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 1 0 0 1 0 0 1 0 0 0
 0 0 0 0 0 0 0 1 0 0 1 0 0 0 1 0 0 0 1 0 0 0 1 0 0 0 0 0 1 1 0 0 1 0 0 0 0 0 0 0 0
 0 1 0 1 0 0 0 0 0 0 1 1 0 1 0 0 1 1 1 1 0 0 0 0 0 0]
El coeficiente del Modelo Ensamblado es  93.0 %
```

Se puede ver que en el bloque de datos de entrenamiento del modelo Ensamblado, nos ofrece un 93.5%, esto representa la validación o robustez del modelo y en el bloque de datos de prueba 93 % de confiabilidad.

En la métrica denominado matriz de confusión, analizaremos la cantidad de elementos que reconoce correctamente; es decir estudiantes que no desertan (0) y estudiantes que si desertor (1).

Figura 68

Matriz de confusión del modelo Ensamblado



Análisis de la Matriz de confusión:

De 72 estudiantes que no desertan en la prueba de estudio, el modelo Ensamblado, reconoce a 70 y se equivoca en 2 estudiantes. Sostiene que 2 si desertan.

De 28 estudiantes que, si desertan en la prueba de estudio, el modelo Ensamblado, reconoce a 23 y no reconoce a 5. Sostiene que 5 no desertan.

Error en la predicción=7 estudiantes.

Conclusión:

*De los 4 modelos analizados 2 de ellos modelo 1 y modelo 2(**Regresión Logística Binariay Naive Bayes**), son los que nos ofrecen los más altos índices de confiabilidad (93.0%), ambos tienen el mismo reconocimiento de patrones. En estudiantes que no desertan, ambos modelos aciertan en 70 de un total de 72 estudiantes de la prueba y en estudiantes que, si desertan, aciertan en 23 de 28 estudiantes de la prueba.*

Error global: 7 estudiantes no identificados correctamente.

En el Modelo Ensamblado, sus resultados coinciden con los resultados del modelo 1 y 2, lo cual confirma la eficiencia de estos.

*Por lo tanto, el **modelo ensamblado**, sería la propuesta de este trabajo de investigación por lo siguiente:*

Además de ofrecer un elevado índice de eficiencia en la predicción de 93%, en algunos casos mayor y otros igual que los modelos básicos, el propósito de los modelos ensamblados no solo es mejorar la eficiencia de los modelos básicos sino mejorar su robustez. Esta característica de calidad en la industria del software, garantiza que el índice de eficiencia se mantenga constante durante todo el ciclo de vida del modelo.

4.1.6.- Implementación:

El presente trabajo de investigación, concluye con la implementación y prueba de un modelo Ensamblado. El modelo Ensamblado es la base para la construcción de soluciones tanto en web, apps o de escritorio, útiles para usuarios finales relacionados el quehacer educativo.

Figura 69

Fase de despliegue del modelo

La implementación de estas soluciones no es el propósito del presente trabajo de investigación, sin embargo, dejo aquí una primera versión de implementación del caso propuesto para posteriores investigaciones.

Figura 70

Ejecución de un programa usando el modelo predictivo Ensamblado, para verificar que un estudiante no desertará

```
# Haciendo una prediccion el Modelo Ensamblado con alumno que No Deserta
print("Haciendo la prediccion para un alumno con caracteristicas [1,22,17,0,0,1000,0]")
a= int(input("Tiene Carga Familiar (Nro hijos) ==> "))
b= int(input("Ingrese su Edad ==> "))
c= int(input("Ingresa tu Promedio ponderado del ciclo pasado ==> "))
d= int(input("Resides cerca ala institucion - Cerca=0; Lejos=1 ==> "))
e= int(input("Ingresa su Situacion Laboral - Sin Trabajo=0; Con Trabajo=1==> "))
f= int(input("Digite su Ingreso Familiar ==> "))
g= int(input("Ingresa su tipo de Vivienda(0=propia, 1=alquilada)==> "))

z=ModeloEnsamblado.predict([[a,b,c,d,e,f,g]])
if(z==0):
  print("El Estudiante tiene el ", accuracy_score(y_test, y_predC)*100, "% de probabilidad de No Desertar")
else:
  print("El Estudiante tiene el ", accuracy_score(y_test, y_predC)*100, "% de probabilidad de Desertar")
```

```
Haciendo la prediccion para un alumno con caracteristicas [1,22,17,0,0,1000,0]
Tiene Carga Familiar (Nro hijos) ==> 1
Ingrese su Edad ==> 22
Ingresa tu Promedio ponderado del ciclo pasado ==> 17
Resides cerca ala institucion - Cerca=0; Lejos=1 ==> 0
Ingresa su Situacion Laboral - Sin Trabajo=0; Con Trabajo=1==> 0
Digite su Ingreso Familiar ==> 1000
Ingresa su tipo de Vivienda(0=propia, 1=alquilada)==> 0
El Estudiante tiene el  93.0 % de probabilidad de No Desertar
```

Figura 71

Ejecución de un programa usando el modelo predictivo Ensamblado, para verificar que un estudiante si desertará

```
[ ]  # Haciendo una prediccion el Modelo Ensamblado con alumno que Si Deserta
     print("Haciendo la prediccion para un alumno con caracteristicas [3,27,12,1,1,800,1]")
     a= int(input("Tiene Carga Familiar (Nro hijos) ==> "))
     b= int(input("Ingrese su Edad ==> "))
     c= int(input("Ingresa tu Promedio ponderado del ciclo pasado ==> "))
     d= int(input("Resides cerca ala institucion - Cerca=0; Lejos=1 ==> "))
     e= int(input("Ingresa su Situacion Laboral - Sin Trabajo=0; Con Trabajo=1==> "))
     f= int(input("Digite su Ingreso Familiar ==> "))
     g= int(input("Ingresa su tipo de Vivienda(0=propia, 1=alquilada)==> "))

     z=ModeloEnsamblado.predict([[a,b,c,d,e,f,g]])
     if(z==0):
       print("El Estudiante tiene el ", accuracy_score(y_test, y_predC)*100, "% de probabilidad de No Desertar")
     else:
       print("El Estudiante tiene el ", accuracy_score(y_test, y_predC)*100, "% de probabilidad de Desertar")
```

```
Haciendo la prediccion para un alumno con caracteristicas [3,27,12,1,1,800,1]
Tiene Carga Familiar (Nro hijos) ==> 3
Ingrese su Edad ==> 27
Ingresa tu Promedio ponderado del ciclo pasado ==> 12
Resides cerca ala institucion - Cerca=0; Lejos=1 ==> 1
Ingresa su Situacion Laboral - Sin Trabajo=0; Con Trabajo=1==> 1
Digite su Ingreso Familiar ==> 800
Ingresa su tipo de Vivienda(0=propia, 1=alquilada)==> 1
El Estudiante tiene el  93.0 % de probabilidad de Desertar
```

Contrastación de la hipótesis

A continuación, vamos a demostrar que un modelo predictivo basado en máquinas de aprendizaje supervisadas nos permite predecir la deserción de estudiantes con un nivel superior al 80 %.

De 4 modelos implementados, validados y evaluados basados en algoritmos de aprendizaje supervisados básicos(Regresión logística, bosques aleatorios, soporte vectorial, y naive bayes), ensamblamos un nuevo modelo mediante un algoritmo de votaciones, el mismo que al ser evaluado en su confiabilidad, nos da un 93%, con una tasa de errores en el reconocimiento de patrones de 7/100 (7%) de modo general entre desertores y no desertores, superando ampliamente lo esperado en el proyecto que fue de 80%.

4.2 DISCUSIÓN

1.- Los resultados obtenidos, concuerdan con Masabanda & Zapata(2019), a través de la selección de su modelo predictivo de deserción estudiantil, basado en técnicas de minería de datos, específicamente J48, Random Forest y Sequential Minimal Optimization y en el presente trabajo de investigación, Regresión Logística Binaria, Bosques aleatorios, Maquinas de Soporte Vectorial y Naive Bayes, como entradas para un modelo ensamblado, los resultados son muy a aproximados: 92% de confiabilidad para el modelo seleccionado por Masabanda y 93% de confiabilidad para el modelo ensamblado propuesto. En nuestro caso, dos de cuatro modelos básicos analizados arrojan el mismo resultado:93% para Regresión Logística Binaria y 93% para Naive Bayes, lo cual garantiza aún más la validez y robustez de nuestro modelo ensamblado. Sin embargo, tenemos una gran diferencia con Masabanda en cuanto al objeto de estudio, para ellos, su propósito es determinar las variables de mayor influencia en la deserción estudiantil y luego realizan el modelo. En el presente caso el propósito está centrado en la determinación un modelo ensamblado para predecir la deserción estudiantil; considerando lógicamente en una de las etapas de la metodología empleada, una selección de las mejores dimensiones que conformaran el set de datos, basados en un análisis de correlación de Pearson, teniendo en cuenta una correlación media moderada de (0.5).

Otra distinción es que, el modelo elegido como ganador en ambos trabajos, aún debe pasar a una etapa de producción para poder ser utilizado por usuarios finales y todavía no es una herramienta que podría ser considerada como apoyo para las autoridades universitarias como lo indican.

2.- Esta investigación con Ávila (2021), concuerda en la utilización de algoritmos de aprendizaje supervisado, utilizados en la misma problemática de deserción estudiantil y nos distingue en el objeto de estudio. Para Ávila es determinar la importancia de estos modelos predictivos en la determinación de los estudiantes, así como la percepción de los docentes en función a esta problemática. En nuestro caso puntualizamos a determinar un modelo ensamblado que garantice con muy buen grado de confiabilidad los estudiantes que desertarían en un periodo de tiempo, indicando en nuestra conclusión que el modelo ensamblado propuesto lo hace con 93% de confiabilidad.

3.- Los resultados obtenidos en la investigación, con García(2019), nos distingue en cuanto a la identificación de características relevantes en la deserción estudiantil, García analiza dimensiones relacionadas con la parte académica, notas, créditos, primera, segunda matricula; es decir un hace un análisis interno, en nuestro trabajo consideramos dimensiones obtenidas de las fichas socioeconómica y solo promedios ponderado de registro técnico como fuente de datos, debido a que en la data obtenida de 10 periodos académicos, se observa que la deserción en su mayoría está en función a variables externas a la parte académico, como carga familiar, trabajo estable, uso de tecnología en casa(internet) para sus estudios, salario, vivienda etc. y demostramos con análisis de correlación la dimensiones que más influencia tienen con la variable predictora y con el uso de cuatro modelos basados en algoritmos de aprendizaje supervisados, ensamblar uno nuevo para mejorar la predicción. Asu vez se puede observar en García, que lo más importante es determinar el factor de mayor influencia a diferencia de nuestro trabajo que es identificar el mejor modelo predictivo.

4.- Con respecto a Mamani (2019), la presente investigación, coincide en el uso de Bosques aleatorios algoritmo de clasificación, sin embargo, nos distingue de Mamani en que utiliza bosques aleatorios y otros algoritmos para identificar los factores de mayor influencia en la problemática de la deserción y en el presente trabajo proponemos un mejor modelo predictivo a un nivel mayor del 80% de confiabilidad, el cual se consigue.

5.- Los resultados obtenidos, con respecto a Pérez & Rojas (2020), ante la misma problemática de deserción estudiantil, Pérez diseña un sistema para predecir la deserción de estudiantes, usando el clasificador de Soporte Vectorial, obtiene un modelo predictivo con buenos resultados. Sin embargo, en el presente trabajo de investigación, dicho algoritmo también es analizado y evaluado y produjo buenos resultados, pero no fue el mejor entre otros que analizamos y evaluamos tales como Regresión Logística Binaria Naive Bayes y bosques aleatorios.

6.- La presente investigación, con Padilla (2019), nos distingue en que, se ensambla un mejor modelo a partir de cuatro elegidos el cual permite hacer predicciones al 93% de confiabilidad, y Padilla, logra determinar un modelo predictivo y demostrar la influencia del modelo con la estimación de deserción e incluso determina que la variable más determinante para la deserción es el promedio ponderado del periodo anterior.

CAPÍTULO V CONCLUSIONES Y RECOMENDACIONES

CONCLUSIONES

1. Comprendida la situación problemática, se logró establecer el juego de datos inicial, basado en 12 características, que se recopilaron de las fichas socioeconómicas.
2. Se logró hacer la limpieza de los datos, eliminando características con valores nulos, datos de diferente tipo y algunos registros incompletos logrando obtener el juego de datos adecuado para el modelo con las características más relevantes según correlación moderada de pearson.
3. Se hizo la separación del set de datos en entrenamiento y prueba y logró implementar cuatro modelos predictivos basados en algoritmos de aprendizaje supervisados los que serán entradas para el modelo propuesto ensamblado.
4. Se logró ensamblar un modelo predictivo a partir de los cuatro modelos anteriores, basado en un clasificador de votación denominado Voting.
5. Se midió el grado de confiabilidad del modelo ensamblado, obteniendo un 93% para el modelo ensamblado. Por lo tanto:

   El modelo propuesto sería el modelo ensamblado con 93% de confiabilidad. Aun cuando este mismo indicador se presenta en dos modelos básicos analizados, se elige el modelo ensamblado por su robustez, el cual es una característica que identifica a los modelos ensamblados y garantiza la permanencia de su índice de confiabilidad durante todo su ciclo de vida.

RECOMENDACIONES

1. Se debe mejorar el formato de la ficha socioeconómica, si fuera posible el registro de datos hacerlo de modo digital con la finalidad de evitar datos perdidos, nulos o incompletos.
2. Se recomienda que el **Modelo Ensamblado**, sea aprobado por las autoridades competentes.
3. El modelo ensamblado, debe incluirse en los documentos de gestión institucional tales como el Plan de Desarrollo Informático para ser incorporado dentro de su planificación de sistemas.
4. El modelo ensamblado dentro de los proyectos planificados, se deriven al área de Desarrollo Informático Institucional para su implementación, prueba y puesta en marcha.
5. Derivar el sistema informático predictivo basado en el modelo ensamblado al área de tutoría o bienestar, según sea el caso para estimar las futuras deserciones y poder tomar medidas preventivas.

REFERENCIAS BIBLIOGRÁFICAS

1. Agudelo Viana, L. G., & Aigneren Aburto, J. M. (2008). Diseños de investigación experimental y no-experimental.
2. Andreas, Müller & Sarah Guido, (2016),Introduction to Machine Learning with Python: A Guide for Data Scientists 1st Edición
3. Ávila-Tomás, J. F., Mayer-Pujadas, M. A., & Quesada-Varela, V. J. (2021). La inteligencia artificial y sus aplicaciones en medicina II: importancia actual y aplicaciones prácticas. *Atención Primaria*, *53*(1), 81-88.
4. Bean, JP (1980). Abandonos y rotación: síntesis y prueba de un modelo causal de deserción estudiantil. *Investigación en educación superior* , *12* , 155-187.
5. Breiman L, Friedman JH, Olshen RA, Stone CJ. Classification and Regression Trees, Wadsworth (New York); 2019
6. Burges, C. J. (1998). A tutorial on support vector machines for pattern recognition. *Data mining and knowledge discovery*, *2*(2), 121-167.
7. Caballero, Martín & Riesco, 2019, Big Data con Python Recolección, almacenamiento y proceso.
8. Cano, E. V., Díaz, V. M., Berea, G. A. M., & Garzón, E. G. (2017). La competencia digital del alumnado universitario de Ciencias Sociales desde una perspectiva de género. Prisma Social: revista de investigación social, (19), 347-367.
9. Comisión Intersectorial de Reinserción Educativa (2006). Programa Intersectorial de Re escolarización: Construyendo alternativas educativas para niños, niñas y adolescentes en situación de vulnerabilidad. Santiago de Chile. Foro Nacional Educación de Calidad Para Todos.
10. Cordera Campos, R., Arruti, F., Peralta, J., Popoca, A., Sheinbaum, D., & Victoria, J. L. (2007). *Temas de la educación superior en América Latina y el Caribe*. UDUAL.
11. Cortés, C. y Vapnik, V. (1995). Redes de vectores de soporte. *Aprendizaje automático* , *20* , 273-297.
12. Cullen, FT y Tinto, V. (1975). Un análisis mertoniano de la desviación escolar.
13. Domínguez Velad, M. (2021). *Autoajuste de hiperparámetros para metaheurísticas* (Doctoral dissertation, ETSI_Informatica).

14. Durkheim, E. (1951). *Suicide [1897]*. na.
15. Escalona, M. B. (2020). *Análisis de datos categóricos: regresión logística y multinomial*
16. Fayyad, UM, Smyth, P., Weir, N. y Djorgovski, S. (1995). Análisis y exploración automatizada de bases de datos de imágenes: resultados, avances y desafíos. *Revista de sistemas de información inteligentes* , *4* , 7-25.
17. Fiuza, D., & Rodriguez, J. (2009). La regresión logística: una herramienta versátil. *Nefrologia*, *20*(6), 477-565. https://www.revistanefrologia.com/es-la-regresion-logistica-una-herramienta-articulo-X0211699500035664
18. García Franco, Jacobo. 2019. "Implementación de un Modelo Computacional basado en Reglas de Clasificación Supervisadas para la Predicción de la Deserción Estudiantil en la Universidad Peruana Unión Filial Juliaca", Universidad Peruana Unión, Sede Juliaca – Perú.
19. García, García, Gonzales y García (2010). *Modelo de regresión logística para estimar la dependencia según la escala de Lawton y Brody*. (s. f.).
20. Gerón Aurélien . 2019. Aprende Machine Learning con Svcikit – Learn, Keras y Tensor Flow.
21. Gironés, Casas, Minguillón & Caihuelas, 2017. Minería de Datos modelos y Algoritmos
22. Gonzalez, L. (2019a, April 5). Bosque Aleatorios Regresión - Teoría - Aprende IA. Aprende IA. https://aprendeia.com/bosques-aleatorios-regresion-teoria-machine-learning/
23. Gonzalez, L. (2019b, September 20). Naive Bayes – Teoría. Aprende IA. https://aprendeia.com/algoritmo-naive-bayes-machine-learning/
24. Himmel, E. (2002). Modelo de análisis de la deserción estudiantil en la educación superior. *Calidad en la Educación* , (17), 91-108.
25. Hu Tao, Fan Xin, Wang Shuo, Guo Zizheng, Liu Aichang, Huang Faming. Landslide susceptibility evaluation of Sinan County using logistics regression model and 3S technology[J]. Bulletin of Geological Science and Technology, 2020, 39(2): 113-121. doi: 10.19509/j.cnki.dzkq.2020.0212
26. Hastie, Trevor, Robert Tibshirani, y Jerome Friedman, 2009: The Elements of Statistical Learning:Data Mining, Inference and Prediction, segunda edición, Nueva York: Springer-Verlag.

27. Heredia, J. J., Rodríguez, A. G., & Vilalta, J. A. (2014). Predicción del rendimiento en una asignatura empleando la regresión logística ordinal. *Estudios pedagógicos (Valdivia)*, *40*(1), 145-162.

28. Hernández-Sampieri, R., Fernández-Collado, C., & Baptista-Lucio, P. (2006). Analisis de los datos cuantitativos. *Metodología de la investigación*, 407-499.

29. Joyanes Aguilar, Luis. 2023. Ciencia de Datos Un enfoque práctico de tecnologías, herramientas y aplicaciones.

30. Lawton MP, Brody EM. Assessment of older people: self-maintaining and instrumental activities of daily living. Gerontologist. 2010;9:179–86

31. López EK, Juárez F, Acevedo M. 2010. Métodos y técnicas de investigación. En: Reidl LM, Mas O (editores). Metodología Científica y Aplicación de la Estadística Descriptiva e Inferencial. México.

32. Lopez Boada, María Jesús, Beatriz Lopez Boada, y Vicente Diaz Lopez, 2005: "Algoritmo de aprendizaje por refuerzo continuo para el control de un sistema de suspensión semi-activa", Revista Iberoamericana de Ingeniería Mecánica, volumen 9, número 2, 77-91.

33. Maguire, D. J., Batty, M., & Goodchild, M. F. (2005). GIS, spatial analysis, and modeling. *(No Title)*.

34. Mamani Padilla, Diego Ismael. (2019). "Modelo de minería de datos basado en factores asociados para la predicción de deserción estudiantil universitaria", Universidad Nacional de Moquegua – Perú.

35. Masabanda Yépez, J. F., & Zapata Rocha, C. J. (2019). *Modelo basado en minería de datos para determinar factores de deserción estudiantil en la facultad de ciencias de la ingeniería y aplicadas de la universidad técnica de Cotopaxi* (Bachelor's thesis, Ecuador: Latacunga: Universidad Técnica de Cotopaxi (UTC).).

36. Manos Antoninis, 2020, Informe de Seguimiento de la Educación en el Mundo América Latina y El Caribe Inclusión y Educación Todas y Todos sin excepción.

37. Moor, James, 2006: "The Dartmouth College Artificial Intelligence Conference: The Next Fifty Years", AI Magazine, volumen 27, número 4, 87.

38. Müller, AC y Guido, S. (2016). *Introducción al aprendizaje automático con Python: una guía para científicos de datos* . "O'Reilly Media, Inc.".

39. Nandeshwar, A., Menzies, T. y Nelson, A. (2011). Patrones de aprendizaje de retención de estudiantes universitarios. *Sistemas expertos con aplicaciones*, *38* (12), 14984-14996.
40. Padilla, R. D. M. (2019). La llegada de la inteligencia artificial a la educación. *Revista de Investigación en Tecnologías de la Información: RITI*, *7*(14), 260-270.
41. Paredes Esparza, R., Francisca, A. L., & Quense Abarzúa, M. D. L. Á. (2017). Modelo de retención universitaria: desafíos y oportunidades en su diseño e implementación.
42. Pereira, R. T. (2010). Una lectura sobre deserción universitaria en estudiantes de pregrado desde la perspectiva de la minería de datos. Revista Guillermo de Ockham, 8(1), 121-130.
43. Pérez Vargas, J. J., Nieto Bravo, J. A., Santamaría Rodríguez, J. E., Moncada Guzmán, C. J., Quintero Torres, F. A., Ortiz Jiménez, J. G., ... & Rojas Mesa, J. E. (2020). Reflexiones metodológicas de investigación educativa:: perspectivas sociales. Ediciones uStA.
44. Pineda, J. M. (2022). Modelos predictivos en salud basados en aprendizaje de maquina (machine learning). Revista Médica Clínica Las Condes, 33(6), 583-590.
45. Ramos, J. R. G. (2018). Cómo se construye el marco teórico de la investigación. Cuadernos de Pesquisa, 48, 830-854. https://www.scielo.br/j/cp/a/xpbhxtDHLrGHfLPthJHQNwK/
46. Russell, S. and P. Norvig, Artificial Intelligence: A Modern Approach. Second ed. Upper Saddle River (N J): Prentice Hall/ Pearson Education; 2003.
47. Reyes Rocabado, J., Escobar Flores, C., Duarte Vargas, J., & Ramirez Peradotto, P. (2007). Una aplicación del modelo de regresión logística en la predicción del rendimiento estudiantil. Estudios pedagógicos (Valdivia), 33(2), 101-120.
48. Salvador Blanco, L., & García-Valcárcel Muñoz-Repiso, A. M. (1989). El rendimiento académico en la Universidad de Cantabria: abandono y retraso en los estudios.
49. Spady, WG (1970). Abandonos de la educación superior: una revisión y síntesis interdisciplinaria. Intercambio , 1 (1), 64-85.

50. Spositto, O. M., Etcheverry, M. E., Ryckeboer, H. L., & Bossero, J. (2010). Aplicación de técnicas de minería de datos para la evaluación del rendimiento académico y la deserción estudiantil (Orlado: Novena Conferencia Iberoamericana en Sistemas, Cibernética e Informática).
51. Stahl, V. V., & Pavel, D. M. (1992). Assessing the Bean and Metzner Model with Community College Student Data.
52. Sutton, Richard, y Andrew Barto, 1998: Reinforcement learning: An introduction, Cambridge: The MIT Press.
53. Tinto, V. (1975). Dropout from higher education: A theoretical synthesis of recent research. Review of educational research, 45(1), 89-125.
54. Tinto, V. (1982). Definición de deserción escolar: una cuestión de perspectiva. Nuevas direcciones para la investigación institucional , 1982 (36), 3-15.

ANEXO 1 Código Programa

```
# Autor : Victor Jaime Polo Romero - 2023
"""Caso3TesisFirme.ipynb
Automatically generated by Colaboratory.
Original file is located at
    https://colab.research.google.com/drive/1Eh8C9C-fhpk0A5860v0HsmLV07TsY0Zf
"""
```

Importando Librerias a utilizar

In [ ]:

```
## importing models
from sklearn.ensemble import RandomForestClassifier
from sklearn.linear_model import LogisticRegression
from sklearn.naive_bayes import GaussianNB
from sklearn.svm import SVC
from sklearn.ensemble import VotingClassifier
from sklearn.metrics import accuracy_score
from sklearn.model_selection import train_test_split
import matplotlib.pyplot as plt
import warnings
warnings.filterwarnings("ignore")
```

In [ ]:

Importando la data Balanceada y Discretizada a un DataFrame(Hoja de calculo - Excel)

In [ ]:

```
import pandas as pd
URL='/content/drive/MyDrive/Colab Notebooks/Caso1Tesis/DataSet_Tesis24.xlsx'
df = pd.read_excel(URL)
df
```

Out[3]:

	Nro	CarFam	EdaEst	PromPon	Proced_Est(0=C;1=L)	Sit_Laboral(0=No;1=Si)	Ingreso_Fam	Vivienda(0=Propia;1=No)	Serv_Internet(0=No;1=Si)	Seguro_Salud(0=Sis;1=Essalud)	Reg_Aliment(0=2Veces;1=3Veces)	Tiene_Discap(0=No;1=Si)	Con_Quien_Vive(0=Solo;1=Familia)	Desertor(0=No;1=Si)
0	1	0	22	18	0	0	1800	0	1	1	1	0	1	0
1	2	2	28	12	1	1	1025	1	0	0	2	0	1	1
2	3	1	21	15	0	0	1300	0	1	1	1	0	1	0
3	4	3	31	13	1	1	1500	1	1	1	1	0	1	1
4	5	1	20	15	0	0	1800	0	1	1	1	0	0	0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
495	496	0	25	14	0	0	2000	0	0	0	1	0	1	1
496	497	0	20	17	0	0	1950	0	0	0	1	0	1	0
497	498	3	27	13	1	1	1200	1	0	0	0	0	1	1
498	499	0	18	17	0	0	1550	0	0	0	1	0	0	0
499	500	0	19	18	0	0	1600	0	0	0	1	0	0	0

500 rows × 14 columns

Limpieza de datos: Eliminando datos perdidos

In [ ]:

```
# Borrando datos si hubieran vacios

df = df.dropna()
```

Verificando la limpieza de los datos

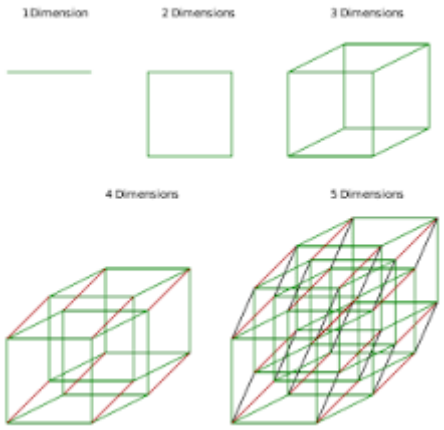
In [ ]:

```
#Verificando Columnas sin datos(Limpieza)
df.isna().sum()
```

Out[5]:

```
Nro                                  0
CarFam                               0
EdaEst                               0
PromPon                              0
Proced_Est(0=C;1=L)                  0
Sit_Laboral  (0=No;1=Si)             0
Ingreso_Fam                          0
Vivienda(0=Propia;1=No)              0
Serv_Internet(0=No;1=Si)             0
Seguro_Salud(0=Sis;1=Essalud)        0
Reg_Aliment(0=2Veces;1=3Veces)       0
Tiene_Discap(0=No;1=Si)              0
Con_Quien_Vive(0=Solo;1=Familia)     0
Desertor(0=No;1=Si)                  0
dtype: int64
```

Verificando las dimensiones del DataSet



In [ ]:

df.shape

Out[6]:

```
(500, 14)
```

Verificando el DataSet

In [ ]:

df

Out[6]:

	Nro	CarFam	EdaEst	PromPon	Proced_Est(0=C;1=L)	Sit_Laboral(0=No;1=Si)	Ingreso_Fam	Vivienda(0=Propia;1=No)	Serv_Internet(0=No;1=Si)	Seguro_Salud(0=Sis;1=Essalud)	Reg_Aliment(0=2Veces;1=3Veces)	Tiene_Discap(0=No;1=Si)	Con_Quien_Vive(0=Solo;1=Familia)	Desertor(0=No;1=Si)
0	1	0	22	18	0	0	1800	0	1	1	1	0	1	0

	Nro	CarFam	EdaEst	PromPon	Proced_Est(0=C;1=L)	Sit_Laboral(0=No;1=Si)	Ingreso_Fam	Vivienda(0=Propia;1=No)	Serv_Internet(0=No;1=Si)	Seguro_Salud(0=Sis;1=Essalud)	Reg_Aliment(0=2Veces;1=3Veces)	Tiene_Discap(0=No;1=Si)	Con_Quien_Vive(0=Solo;1=Familia)	Desertor(0=No;1=Si)
1	2	2	28	12	1	1	1025	1	0	0	2	0	1	1
2	3	1	21	15	0	0	1300	0	1	1	1	0	1	0
3	4	3	31	13	1	1	1500	1	1	1	1	0	1	1
4	5	1	20	15	0	0	1800	0	1	1	1	0	0	0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
495	496	0	25	14	0	0	2000	0	0	0	1	0	1	1
496	497	0	20	17	0	0	1950	0	0	0	1	0	1	0
497	498	3	27	13	1	1	1200	1	0	0	0	0	1	1
498	499	0	18	17	0	0	1550	0	0	0	1	0	0	0
499	500	0	19	18	0	0	1600	0	0	0	1	0	0	0

500 rows × 14 columns

Eliminando la columna Nro por ser irrelevante para el modelo

In [ ]:

```
# Remover Columna "Nro" que no contribuye al modelo
df = df.drop('Nro', axis=1)
df.columns
```

Visualizando el DataSet con la columna Nro eliminada

In [ ]:

```
# Verificando las dimensiones y visualizando el nuevo df
rows=df.shape[0]
columns=df.shape[1]
print ( rows,columns )
df
```

```
500 13
```

Out[9]:

	CarFam	EdaEst	PromPon	Proced_Est(0=C;1=L)	Sit_Laboral(0=No;1=Si)	Ingreso_Fam	Vivienda(0=Propia;1=No)	Serv_Internet(0=No;1=Si)	Seguro_Salud(0=Sis;1=Essalud)	Reg_Aliment(0=2Veces;1=3Veces)	Tiene_Disap(0=No;1=Si)	Con_Quien_Vive(0=Solo;1=Familia)	Desertor(0=No;1=Si)
0	0	22	18	0	0	1800	0	1	1	1	0	1	0
1	2	28	12	1	1	1025	1	0	0	2	0	1	1
2	1	21	15	0	0	1300	0	1	1	1	0	1	0
3	3	31	13	1	1	1500	1	1	1	1	0	1	1
4	1	20	15	0	0	1800	0	1	1	1	0	0	0
...	...	...	...	...	...	...	...	...	...	...	...	...	...
495	0	25	14	0	0	2000	0	0	0	1	0	1	1
496	0	20	17	0	0	1950	0	0	0	1	0	1	0
497	3	27	13	1	1	1200	1	0	0	0	0	1	1
498	0	18	17	0	0	1550	0	0	0	1	0	0	0
499	0	19	18	0	0	1600	0	0	0	1	0	0	0

500 rows × 13 columns

In [ ]:

df.head()

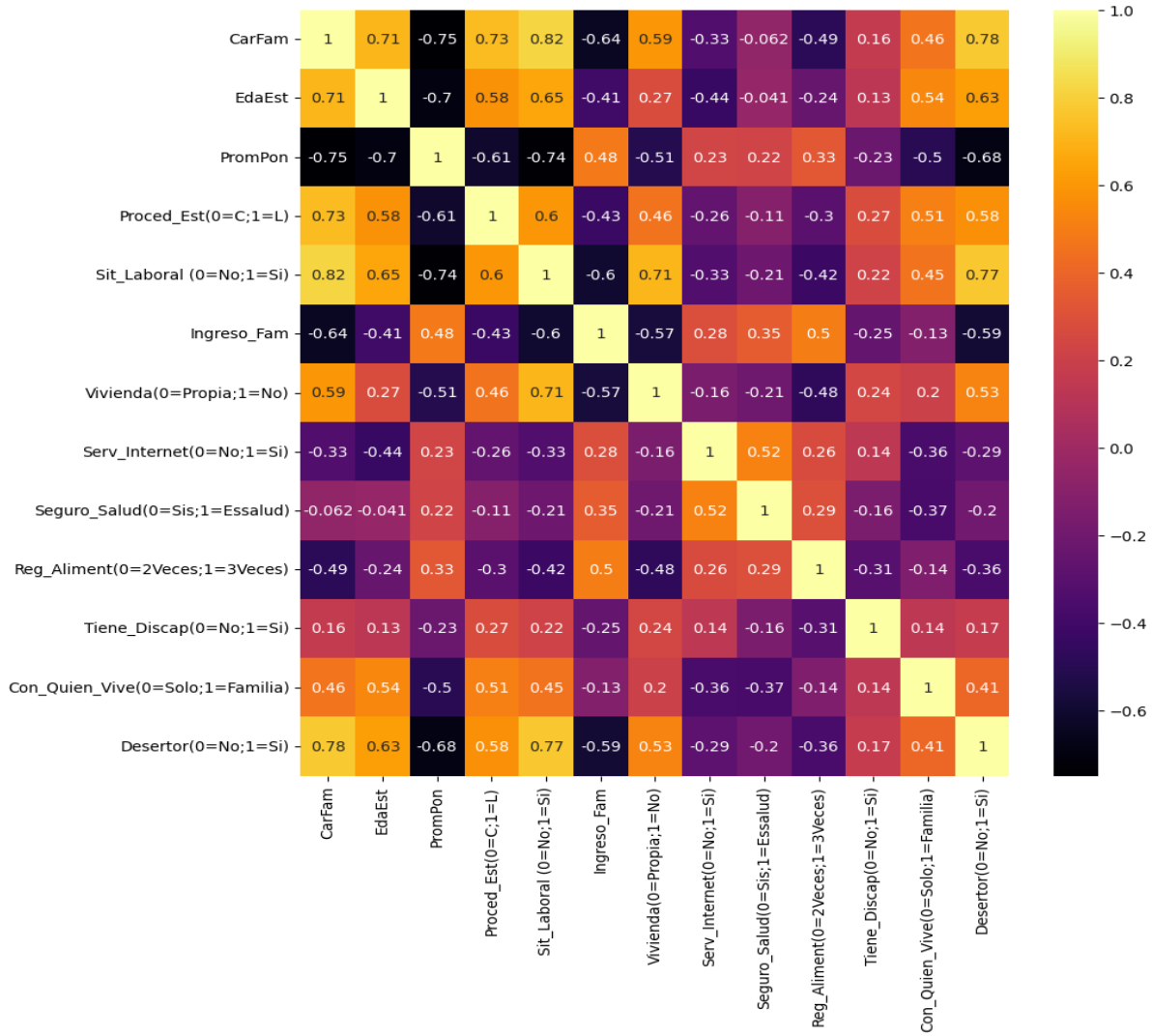
Out[10]:

	CarFam	EdaEst	PromPon	Proced_Est(0=C;1=L)	Sit_Laboral(0=No;1=Si)	Ingreso_Fam	Vivienda(0=Propia;1=No)	Serv_Internet(0=No;1=Si)	Seguro_Salud(0=Sis;1=Essalud)	Reg_Aliment(0=2Veces;1=3Veces)	Tiene_Discap(0=No;1=Si)	Con_Quien_Vive(0=Solo;1=Familia)	Desertor(0=No;1=Si)
0	0	22	18	0	0	1800	0	1	1	1	0	1	0
1	2	28	12	1	1	1025	1	0	0	2	0	1	1
2	1	21	15	0	0	1300	0	1	1	1	0	1	0
3	3	31	13	1	1	1500	1	1	1	1	0	1	1
4	1	20	15	0	0	1800	0	1	1	1	0	0	0

Reduciendo la dimensionalidad con datos irrelevantes

In [ ]:

```
# Se usara como criterio: El grado de correlacion entre las dimesiones de la variable X y la varable predictiva Y(Desertor)
#Para Reducir la dimensionalidad
corr = df.corr()
# Usando Seaborn y creando un Mapa de Calor
import seaborn as sns
plt.figure(figsize=(10,10))
sns.heatmap(corr, annot = True, cmap='inferno')
plt.show()
```



In [ ]:

Seleccionando columnas donde la correlacion entre las variables dependientes e indepenidente sea mayor a 0.5

In [ ]:

```
# Seleccionar las columnas con mayor correlacion  respecto a la Desercion.
corr[abs(corr['Desertor(0=No;1=Si)']) > 0.5]
```

Out[12]:

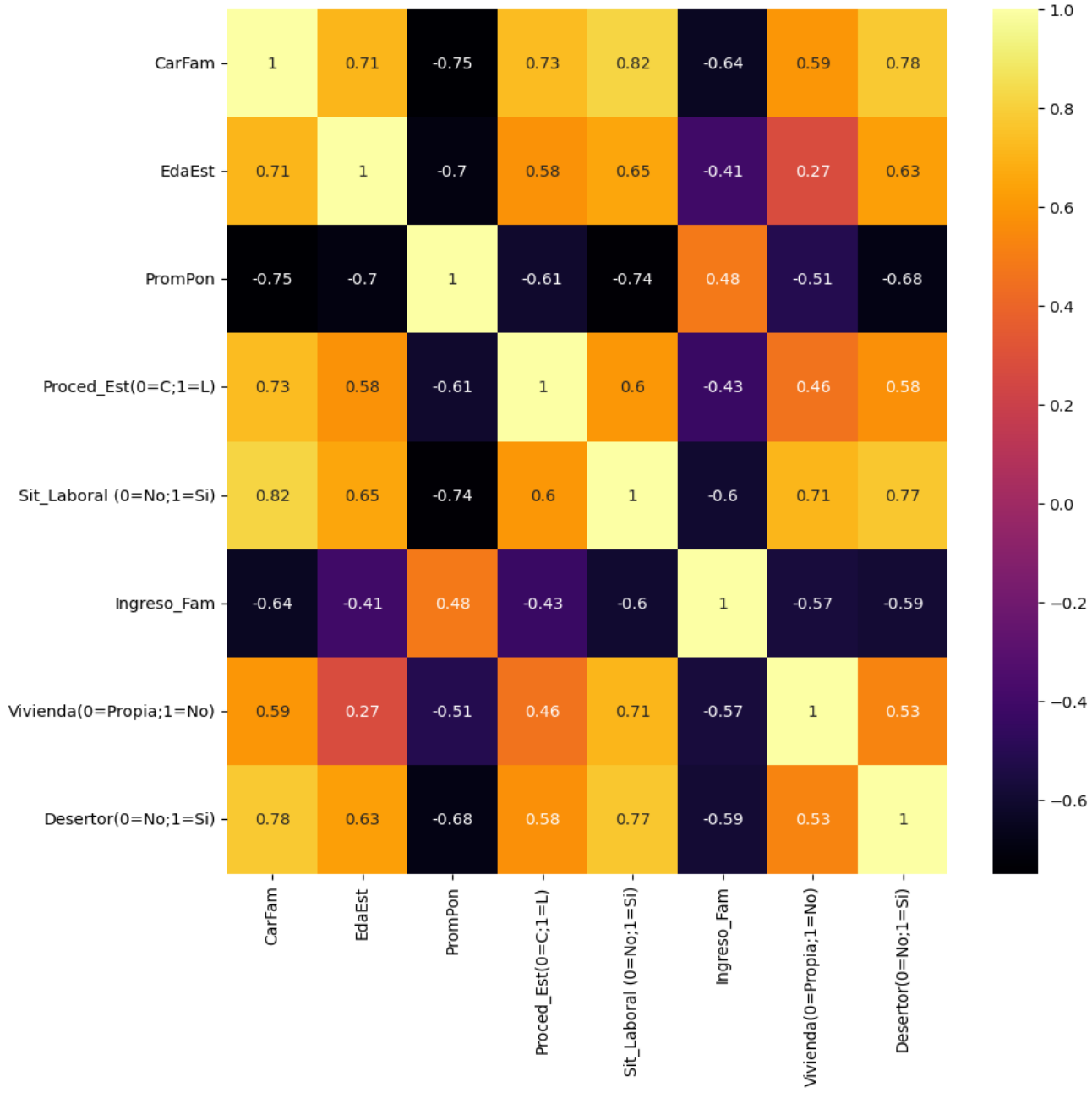
	CarFam	EdaEst	PromPon	Proced_Est(0=C;1=L)	Sit_Laboral(0=No;1=Si)	Ingreso_Fam	Vivienda(0=Propia;1=No)	Serv_Internet(0=No;1=Si)	Seguro_Salud(0=Sis;1=Essalud)	Reg_Aliment(0=2Veces;1=3Veces)	Tiene_Discap(0=No;1=Si)	Con_Quien_Vive(0=Solo;1=Familia)	Desertor(0=No;1=Si)
CarFam	1.000000	0.712805	-0.749520	0.733252	0.822167	-0.642793	0.590623	-0.333199	-0.062426	-0.486720	0.162142	0.460132	0.780689
EdaEst	0.712805	1.000000	-0.700666	0.578953	0.653693	-0.405781	0.270281	-0.439021	-0.040679	-0.238353	0.128253	0.541152	0.627928
PromPon	-0.749520	-0.700666	1.000000	-0.607012	-0.740259	0.481919	-0.512677	0.228080	0.222256	0.333277	-0.228645	-0.497263	-0.683085
Proced_Est(0=C;1=L)	0.733252	0.578953	-0.607012	1.000000	0.600533	-0.431427	0.455690	-0.262856	-0.112826	-0.296554	0.268991	0.510250	0.575684
Sit_Laboral(0=No;1=Si)	0.822167	0.653693	-0.740259	0.600533	1.000000	-0.599748	0.706441	-0.331957	-0.211014	-0.424994	0.218218	0.454257	0.772507
Ingreso_Fam	-0.642793	-0.405781	0.481911	-0.431427	-0.599748	1.000000	-0.565717	0.283797	0.351980	0.496045	-0.254782	-0.125967	-0.586454

	CarFam	EdaEst	PromPon	Proced_Est(0=C;1=L)	Sit_Laboral (0=No; 1=Si)	Ingreso_Fam	Vivienda(0=Propia;1=No)	Serv_Internet(0=No; 1=Si)	Seguro_Salud (0=Sis; 1=Essalud)	Reg_Aliment(0=2Veces;1=3Veces)	Tiene_Discap(0=No; 1=Si)	Con_Quien_Vive(0=Solo;1=Familia)	Desertor(0=No; 1=Si)
	93	81	99										
Vivienda(0=Propia;1=No)	0.590623	0.270281	-0.512677	0.455690	0.706441	-0.565717	1.000000	-0.160627	-0.209431	-0.475885	0.241008	0.204434	0.527800
Desertor(0=No; 1=Si)	0.780689	0.627928	-0.683085	0.575684	0.772507	-0.586454	0.527800	-0.288437	-0.196413	-0.362706	0.166449	0.408529	1.000000

Grafico de calor con dimensiones significativas, segun Pearson > 0.5

In [ ]:

```
#Nuevo Grfico de calor con dimesiones significativas segun pearson >0.5 moderadamente significativas
car = ['CarFam', 'EdaEst', 'PromPon', 'Proced_Est(0=C;1=L)',
    'Sit_Laboral (0=No;1=Si)', 'Ingreso_Fam','Vivienda(0=Propia;1=No)','Desertor(0=No;1=Si)']
df1=df[car]
corr = df1.corr()
# Usando Seaborn y creando un Mapa de Calor
import seaborn as sns
plt.figure(figsize=(10,10))
sns.heatmap(corr, annot = True, cmap='inferno')
plt.show()
```



Set de Datos con el que se realizara los Modelos

In [ ]:

 # El Set de Datos con el que se trabajara

df1

Out[14]:

	CarFam	EdaEst	PromPon	Proced_Est(0=C;1=L)	Sit_Laboral (0=No;1=Si)	Ingreso_Fam	Vivienda(0=Propia;1=No)	Desertor(0=No;1=Si)
0	0	22	18	0	0	1800	0	0
1	2	28	12	1	1	1025	1	1
2	1	21	15	0	0	1300	0	0

	CarFam	EdaEst	PromPon	Proced_Est(0=C;1=L)	Sit_Laboral (0=No;1=Si)	Ingreso_Fam	Vivienda(0=Propia;1=No)	Desertor(0=No;1=Si)
3	3	31	13	1	1	1500	1	1
4	1	20	15	0	0	1800	0	0
...	...	...	...	...	...	...	...	...
495	0	25	14	0	0	2000	0	1
496	0	20	17	0	0	1950	0	0
497	3	27	13	1	1	1200	1	1
498	0	18	17	0	0	1550	0	0
499	0	19	18	0	0	1600	0	0

500 rows × 8 columns

##Separando el Set de datos en Y para la variable predictiva y X para las dimensioens de la variable X

In [ ]:

```
##Separando el Set de datos en Y para la variable predictiva y X para las dimensioens de la variable X
## Entrada: Variables = x
## Salida: variables = y
y=df1['Desertor(0=No;1=Si)']
x=df1.drop('Desertor(0=No;1=Si)',axis=1)
x
```

Out[15]:

	CarFam	EdaEst	PromPon	Proced_Est(0=C;1=L)	Sit_Laboral (0=No;1=Si)	Ingreso_Fam	Vivienda(0=Propia;1=No)
0	0	22	18	0	0	1800	0
1	2	28	12	1	1	1025	1
2	1	21	15	0	0	1300	0
3	3	31	13	1	1	1500	1
4	1	20	15	0	0	1800	0
...	...	...	...	...	...	...	...
495	0	25	14	0	0	2000	0

	CarFam	EdaEst	PromPon	Proced_Est(0=C;1=L)	Sit_Laboral (0=No;1=Si)	Ingreso_Fam	Vivienda(0=Propia;1=No)
496	0	20	17	0	0	1950	0
497	3	27	13	1	1	1200	1
498	0	18	17	0	0	1550	0
499	0	19	18	0	0	1600	0

500 rows × 7 columns

In [ ]:

```
#Visualizando y
y
```

Out[16]:

```
0       0
1       1
2       0
3       1
4       0
       ..
495     1
496     0
497     1
498     0
499     0
Name: Desertor(0=No;1=Si), Length: 500, dtype: int64
```

In [ ]:

```
# PASAMOS DE 13 COLUMNAS A 8 COLUMNAS

df1.shape
```

Out[17]:

```
(500, 8)
```

In [ ]:

```
## ESCOGER 'X' (Caracteristicas) y 'y' (TARGET)

# Usando SOLO las 7 Columnas seleccionadas (columnas con Correlación mayor a 0.5 con la columna de "Desertor")

X = x

y =y

X
```

Out[18]:

	CarFam	EdaEst	PromPon	Proced_Est(0=C;1=L)	Sit_Laboral (0=No;1=Si)	Ingreso_Fam	Vivienda(0=Propia;1=No)
0	0	22	18	0	0	1800	0
1	2	28	12	1	1	1025	1
2	1	21	15	0	0	1300	0
3	3	31	13	1	1	1500	1
4	1	20	15	0	0	1800	0
...	...	...	...	...	...	...	...
495	0	25	14	0	0	2000	0
496	0	20	17	0	0	1950	0
497	3	27	13	1	1	1200	1
498	0	18	17	0	0	1550	0
499	0	19	18	0	0	1600	0

500 rows × 7 columns

In [ ]:

```
#Verificando la variable desertor
y
```

Out[19]:

```
0      0
1      1
2      0
3      1
4      0
      ..
495    1
496    0
497    1
498    0
499    0
Name: Desertor(0=No;1=Si), Length: 500, dtype: int64
```

Separando el Set de Datos en Entrenamiento(80%) y Prueba(20%)

In [ ]:

```
## Separando nuestra data en entrenamiento y prueba

X_train,X_test,y_train, y_test = train_test_split( X,y,test_size=.20,random_state=42323232)
```

Aqui se Crean los modelos basados en los algoritmos de aprendizaje

In [ ]:

```
## Instanciando los modelos - Paso 10
Algoritmo1 = LogisticRegression()
Algoritmo2 = GaussianNB()
Algoritmo3 = RandomForestClassifier(n_estimators=100)
Algoritmo4 = SVC()
```

In [ ]:

Librerias para la Matriz de Confusion

In [ ]:

```
# Librerias para Matriz de Condusion
from mlxtend.plotting import plot_confusion_matrix
from sklearn.metrics import confusion_matrix
```

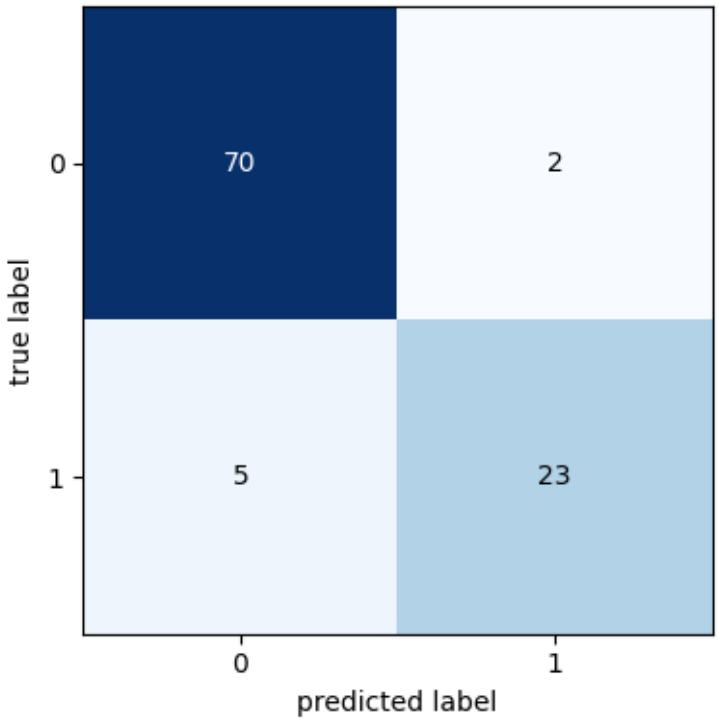
Entrenando los algoritmos con el Set de Datos. Haciendo predicciones para validar el Modelo y Evaluar la Confiabilidad.

* Modelo 1 basado en Regresion Logistica Binaria*

In [ ]:

```
# Entrenando el Algoritmo1
Modelo1=Algoritmo1.fit(X_train, y_train)

# Validando el Modelo1 con una prediccion de validacion
y_predV = Modelo1.predict(X_train)
print("El Coeficiente de Validacion del Modelo1 basado en ", Algoritmo1, "es : ",accuracy_score(y_tr
ain, y_predV)*100 ,"%")

# Evaluando la Confiabilidad del Modelo1 con una prediccion de evaluacion
y_predC = Modelo1.predict(X_test)
print("El Coeficiente de Confiabilidad del Modelo1 basado en ", Algoritmo1, " es : ",accuracy_score(y
_test, y_predC)*100 ,"%")

print("Datos del Vector Prueba \n",y_test)
print("Datos del Vector Prediccion\n",y_predC)
# Matriz en modo texto
matriz = confusion_matrix(y_test,y_predC)
plot_confusion_matrix(conf_mat=matriz, figsize=(4,4), show_normed=False)
plt.tight_layout()
```

```
El Coeficiente de Validacion del Modelo1 basado en  LogisticRegression
() es :  93.5 %
El Coeficiente de Confiabilidad del Modelo1 basado en  LogisticRegress
ion()  es :  93.0 %
Datos del Vector Prueba
 106    0
32     1
60     0
337    0
429    1
      ..
160    0
167    0
292    0
427    0
294    0
Name: Desertor(0=No;1=Si), Length: 100, dtype: int64
Datos del Vector Prediccion
 [0 1 0 0 0 1 1 0 0 0 0 1 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 1 0 0 1 0 0 1
0 0 0
 0 0 0 0 0 0 0 1 0 0 1 0 0 0 1 0 0 0 1 0 0 0 0 0 1 1 0 0 1 0 0 0 0 0 0
0 0
 0 1 0 1 0 0 0 0 0 0 1 1 0 1 0 0 1 1 1 1 0 0 0 0 0 0]
```



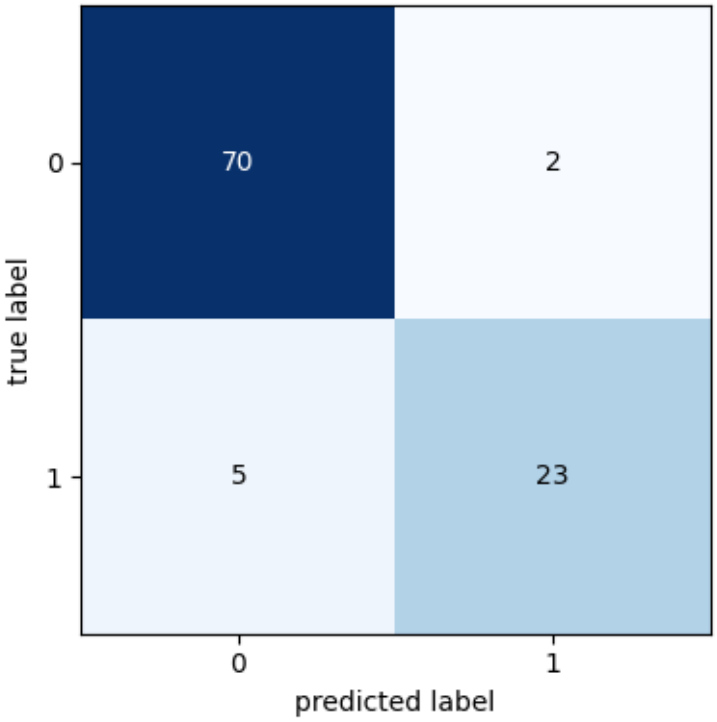
In [ ]:

* Modelo 2 basado en Naive Bayes*

In [ ]:

```
# Entrenando el Algoritmo2
Modelo2=Algoritmo2.fit(X_train, y_train)

# Validando el Modelo1 con una prediccion de validacion
y_predV = Modelo2.predict(X_train)
print("El Coeficiente de Validacion del Modelo2 basado en ", Algoritmo2, "es : ",accuracy_score(y_train, y_predV)*100 ,"%")

# Evaluando la Confiabilidad del Modelo1 con una prediccion de evaluacion
y_predC = Modelo2.predict(X_test)
print("El Coeficiente de Confiabilidad del Modelo1 basado en ", Algoritmo2, " es : ",accuracy_score(y_test, y_predC)*100 ,"%")

print("Datos del Vector Prueba \n",y_test)
print("Datos del Vector Prediccion\n",y_predC)
# Matriz en modo texto
matriz = confusion_matrix(y_test,y_predC)
plot_confusion_matrix(conf_mat=matriz, figsize=(4,4), show_normed=False)
plt.tight_layout()
```

```
El Coeficiente de Validacion del Modelo2 basado en  GaussianNB() es : 
93.5 %
El Coeficiente de Confiabilidad del Modelo1 basado en  GaussianNB()  e
s :  93.0 %
Datos del Vector Prueba
 106    0
32     1
60     0
337    0
429    1
      ..
160    0
167    0
292    0
427    0
294    0
Name: Desertor(0=No;1=Si), Length: 100, dtype: int64
Datos del Vector Prediccion
 [0 1 0 0 0 1 1 0 0 0 0 1 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 1 0 0 1 0 0 1
0 0 0
 0 0 0 0 0 0 0 1 0 0 1 0 0 0 1 0 0 0 1 0 0 0 0 0 1 1 0 0 1 0 0 0 0 0 0
0 0
 0 1 0 1 0 0 0 0 0 0 1 1 0 1 0 0 1 1 1 1 0 0 0 0 0 0]
```

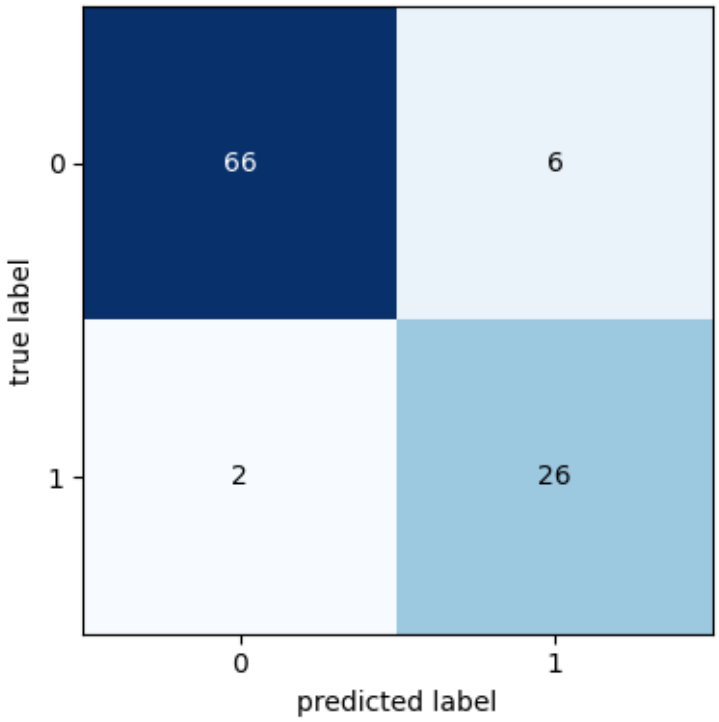


###* Modelo 3 basado en Bosques Aleatorios*

In [ ]:

```
# Entrenando el Algoritmo3
Modelo3=Algoritmo3.fit(X_train, y_train)

# Validando el Modelo1 con una prediccion de validacion
y_predV = Modelo3.predict(X_train)
print("El Coeficiente de Validacion del Modelo3 basado en ", Algoritmo3, "es : ",accuracy_score(y_train, y_predV)*100 ,"%")

# Evaluando la Confiabilidad del Modelo1 con una prediccion de evaluacion
y_predC = Modelo3.predict(X_test)
print("El Coeficiente de Confiabilidad del Modelo1 basado en ", Algoritmo3, " es : ",accuracy_score(y_test, y_predC)*100 ,"%")

print("Datos del Vector Prueba \n",y_test)
print("Datos del Vector Prediccion\n",y_predC)
# Matriz en modo texto
matriz = confusion_matrix(y_test,y_predC)
plot_confusion_matrix(conf_mat=matriz, figsize=(4,4), show_normed=False)
plt.tight_layout()
El Coeficiente de Validacion del Modelo3 basado en  RandomForestClassifier() es :  93.75 %
El Coeficiente de Confiabilidad del Modelo1 basado en  RandomForestClassifier()  es :  92.0 %
Datos del Vector Prueba
 106    0
32     1
```

```
60      0
337     0
429     1
       ..
160     0
167     0
292     0
427     0
294     0
Name: Desertor(0=No;1=Si), Length: 100, dtype: int64
Datos del Vector Prediccion
 [0 1 0 0 1 1 1 0 0 0 0 1 1 0 0 1 1 1 0 0 1 0 0 1 0 0 0 1 0 0 1 0 0 1
0 0 0
 0 0 0 0 0 0 0 1 0 0 1 0 0 0 1 0 0 0 1 1 0 0 0 0 1 1 0 0 1 0 0 1 0 0 0
0 0
 0 1 0 1 0 0 0 0 0 0 1 1 0 1 0 0 1 1 1 1 0 0 0 0 0 0]
```



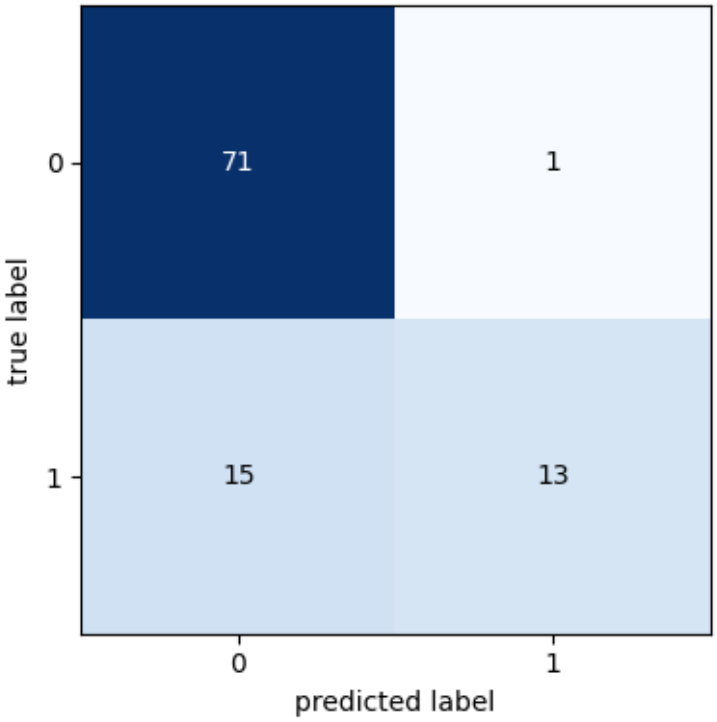
###* Modelo 4 basado en Maquinas de Vectores de Soporte*

In [ ]:

```
# Entrenando el Algoritmo4
Modelo4=Algoritmo4.fit(X_train, y_train)

# Validando el Modelo1 con una prediccion de validacion
y_predV = Modelo4.predict(X_train)
print("El Coeficiente de Validacion del Modelo4 basado en ", Algoritmo4, "es : ",accuracy_score(y_tr
ain, y_predV)*100 ,"%")
```

```
# Evaluando la Confiabilidad del Modelo1 con una prediccion de evaluacion
y_predC = Modelo4.predict(X_test)
print("El Coeficiente de Confiabilidad del Modelo4 basado en ", Algoritmo4, " es : ",accuracy_score(y
_test, y_predC)*100 ,"%")

print("Datos del Vector Prueba \n",y_test)
print("Datos del Vector Prediccion\n",y_predC)
# Matriz en modo texto
matriz = confusion_matrix(y_test,y_predC)
plot_confusion_matrix(conf_mat=matriz, figsize=(4,4), show_normed=False)
plt.tight_layout()
```

```
El Coeficiente de Validacion del Modelo4 basado en  SVC() es :  83.75
%
El Coeficiente de Confiabilidad del Modelo4 basado en  SVC()  es :  84
.0 %
Datos del Vector Prueba
 106    0
32     1
60     0
337    0
429    1
      ..
160    0
167    0
292    0
427    0
294    0
Name: Desertor(0=No;1=Si), Length: 100, dtype: int64
Datos del Vector Prediccion
 [0 1 0 0 0 0 1 0 0 0 0 1 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0
0 0 0
 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0
0 0
 0 0 0 1 0 0 0 0 0 0 1 0 0 1 0 0 1 0 1 0 0 0 0 0 0 0]
```



In [ ]:

```
##RESUMEN
## Detenemos un momento - Paso 11 - OTRA FORMA
for Modelo in (Algoritmo1, Algoritmo2, Algoritmo3, Algoritmo4):
  Modelo.fit(X_train, y_train)

  y_pred = Modelo.predict(X_test)

  print("Modelo Basado en ",Modelo, accuracy_score(y_test, y_pred)*100 ,"%")
```

```
Modelo Basado en  LogisticRegression() 93.0 %
Modelo Basado en  GaussianNB() 93.0 %
Modelo Basado en  RandomForestClassifier() 92.0 %
Modelo Basado en  SVC() 84.0 %
```

###* Modelo Ensamblado a partir del Modelo1, Modelo2, Modelo3, Modelo4*

In [ ]:

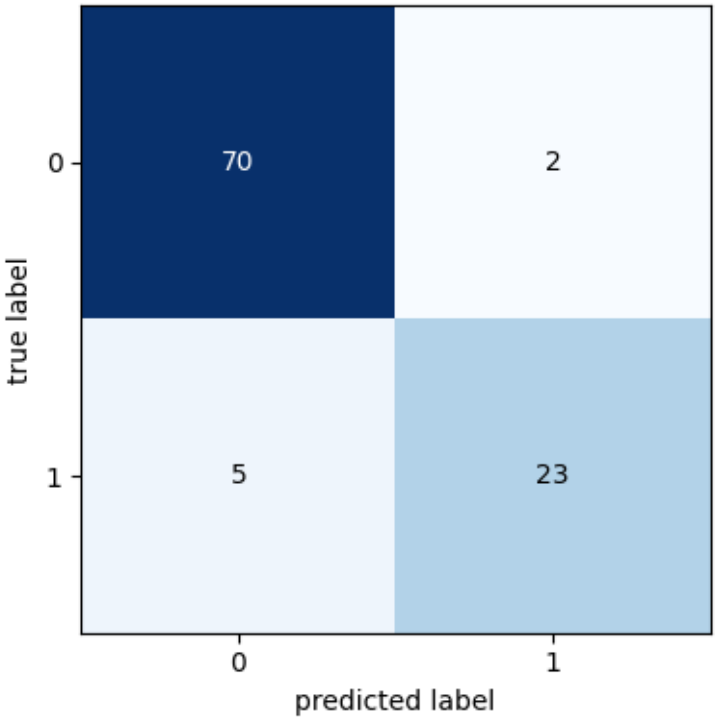
```
## Ensamblando el modelo y haciendo una prueba ok?

ModeloEnsamblado = VotingClassifier(estimators=[('lr', Modelo1), ('nb',Modelo2), ('rf', Modelo3),('
svm', Modelo4)])
ModeloEnsamblado.fit(X_train, y_train)
y_predV = ModeloEnsamblado.predict(X_train)
print("El Modelo Ensamblado esta validado al ", accuracy_score(y_train, y_predV)*100 ,"%")
y_predC = ModeloEnsamblado.predict(X_test)
print("Datos de la Prediccion de Prueba es \n", y_predC)
print("Datos de Originales de la Clase es \n")
for s in y_test:
  print([s])
print("El coeficiente del Modelo Ensamblado es ", accuracy_score(y_test, y_predC)*100 ,"%")
```

```
# Matriz en modo texto
matriz = confusion_matrix(y_test,y_predC)
plot_confusion_matrix(conf_mat=matriz, figsize=(4,4), show_normed=False)
plt.tight_layout()
El Modelo Ensamblado esta validado al  93.5 %
Datos de la Prediccion de Prueba es
 [0 1 0 0 0 1 1 0 0 0 0 1 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 1 0 0 1 0 0 1
0 0 0
 0 0 0 0 0 0 0 1 0 0 1 0 0 0 1 0 0 0 1 0 0 0 0 0 1 1 0 0 1 0 0 0 0 0 0
0 0
 0 1 0 1 0 0 0 0 0 0 1 1 0 1 0 0 1 1 1 1 0 0 0 0 0 0]
Datos de Originales de la Clase es

[0]
[1]
[0]
[0]
[1]
[1]
[1]
[0]
[0]
[0]
[0]
[1]
[1]
[0]
[0]
[1]
[1]
[0]
[0]
[0]
[0]
[0]
[0]
[0]
[1]
[0]
[0]
[0]
[1]
[0]
[0]
[1]
[0]
[0]
[1]
```

[0]
[0]
[0]
[0]
[0]
[0]
[0]
[0]
[0]
[0]
[0]
[0]
[0]
[1]
[0]
[0]
[0]
[1]
[0]
[1]
[0]
[0]
[0]
[0]
[0]
[0]
[0]
[1]
[0]
[0]
[1]
[0]
[0]
[0]
[0]
[0]
[0]
[0]
[0]
[0]
[0]
[0]
[0]
[1]
[0]
[1]
[0]
[1]
[0]
[0]

```
[0]
[0]
[1]
[1]
[0]
[1]
[0]
[0]
[1]
[1]
[1]
[1]
[0]
[0]
[0]
[0]
[0]
[0]
```

El coeficiente del Modelo Ensamblado es 93.0 %



Probando una Implementación con el Modelo Ensamblado

In [ ]:

```
# Haciendo una prediccion el Modelo Ensamblado con alumno que No Deserta
print("Haciendo la prediccion para un alumno con caracteristicas [1,22,17,0,0,1000,0]")
a= int(input("Tiene Carga Familiar (Nro hijos) ==> "))
b= int(input("Ingrese su Edad ==> "))
c= int(input("Ingresa tu Promedio ponderado del ciclo pasado ==> "))
d= int(input("Resides cerca ala institucion - Cerca=0; Lejos=1 ==> "))
e= int(input("Ingresa su Situacion Laboral - Sin Trabajo=0; Con Trabajo=1==> "))
f= int(input("Digite su Ingreso Familiar ==> "))
g= int(input("Ingresa su tipo de Vivienda(0=propia, 1=alquilada)==> "))

z=ModeloEnsamblado.predict([[a,b,c,d,e,f,g]])
if(z==0):
 print("El Estudiante tiene el ", accuracy_score(y_test, y_predC)*100, "% de probabilidad de No Des
ertar")
else:
 print("El Estudiante tiene el ", accuracy_score(y_test, y_predC)*100, "% de probabilidad de Desert
ar")
```

```
Haciendo la prediccion para un alumno con caracteristicas [1,22,17,0,0
,1000,0]
Tiene Carga Familiar (Nro hijos) ==> 1
Ingrese su Edad ==> 22
Ingresa tu Promedio ponderado del ciclo pasado ==> 17
Resides cerca ala institucion - Cerca=0; Lejos=1 ==> 0
Ingresa su Situacion Laboral - Sin Trabajo=0; Con Trabajo=1==> 0
Digite su Ingreso Familiar ==> 1000
Ingresa su tipo de Vivienda(0=propia, 1=alquilada)==> 0
El Estudiante tiene el  93.0 % de probabilidad de No Desertar
```

In [ ]:

```
# Haciendo una prediccion el Modelo Ensamblado con alumno que Si Deserta
print("Haciendo la prediccion para un alumno con caracteristicas [3,27,12,1,1,800,1]")
a= int(input("Tiene Carga Familiar (Nro hijos) ==> "))
b= int(input("Ingrese su Edad ==> "))
c= int(input("Ingresa tu Promedio ponderado del ciclo pasado ==> "))
d= int(input("Resides cerca ala institucion - Cerca=0; Lejos=1 ==> "))
e= int(input("Ingresa su Situacion Laboral - Sin Trabajo=0; Con Trabajo=1==> "))
f= int(input("Digite su Ingreso Familiar ==> "))
g= int(input("Ingresa su tipo de Vivienda(0=propia, 1=alquilada)==> "))

z=ModeloEnsamblado.predict([[a,b,c,d,e,f,g]])
if(z==0):
 print("El Estudiante tiene el ", accuracy_score(y_test, y_predC)*100, "% de probabilidad de No Des
ertar")
else:
 print("El Estudiante tiene el ", accuracy_score(y_test, y_predC)*100, "% de probabilidad de Desert
ar")
```

```
Haciendo la prediccion para un alumno con caracteristicas [3,27,12,1,1
,800,1]
Tiene Carga Familiar (Nro hijos) ==> 3
Ingrese su Edad ==> 27
Ingresa tu Promedio ponderado del ciclo pasado ==> 12
Resides cerca ala institucion - Cerca=0; Lejos=1 ==> 1
Ingresa su Situacion Laboral - Sin Trabajo=0; Con Trabajo=1==> 1
Digite su Ingreso Familiar ==> 800
Ingresa su tipo de Vivienda(0=propia, 1=alquilada)==> 1
El Estudiante tiene el  93.0 % de probabilidad de Desertar
```

In [ ]:

```
print("Los coeficientes del modelo son ", Modelo1.coef_)
print("El Intercepto ", Modelo1.intercept_)
```

```
Los coeficientes del modelo son  [[ 9.74097732e-01  6.18119786e-02 -2.
15208634e-01 -4.02188794e-01
   1.65219620e+00 -7.80500228e-04 -3.21262203e-01]]
El Intercepto  [0.4536182]
```

In [ ]:

E

*** Si bien es cierto que el modelo ensamblado ofrece un mismo porcentaje de confiabilidad que los modelos 1 y 2, debemos resaltar que el modelo ensamblado ofrece mayor robustez que los anteriores.***

ANEXO 2 Predicción con data real al azar

```
# Haciendo una prediccion el Modelo Ensamblado con alumno que No Deserta
print("Haciendo la prediccion para un alumno con caracteristicas
[1,22,17,0,0,1000,0]")
a= int(input("Tiene Carga Familiar (Nro hijos) ==> "))
b= int(input("Ingrese su Edad ==> "))
c= int(input("Ingresa tu Promedio ponderado del ciclo pasado ==> "))
d= int(input("Resides cerca ala institucion - Cerca=0; Lejos=1 ==> "))
e= int(input("Ingresa su Situacion Laboral - Sin Trabajo=0; Con
Trabajo=1==> "))
f= int(input("Digite su Ingreso Familiar ==> "))
g= int(input("Ingresa su tipo de Vivienda(0=propia, 1=alquilada)==> "))

z=ModeloEnsamblado.predict([[a,b,c,d,e,f,g]])
if(z==0):
  print("El Estudiante tiene el ", accuracy_score(y_test, y_predC)*100,
"% de probabilidad de No Desertar")
else:
  print("El Estudiante tiene el ", accuracy_score(y_test, y_predC)*100,
"% de probabilidad de Desertar")

# Haciendo una prediccion el Modelo Ensamblado con alumno que Si Deserta
print("Haciendo la prediccion para un alumno con caracteristicas
[3,27,12,1,1,800,1]")
a= int(input("Tiene Carga Familiar (Nro hijos) ==> "))
b= int(input("Ingrese su Edad ==> "))
c= int(input("Ingresa tu Promedio ponderado del ciclo pasado ==> "))
d= int(input("Resides cerca ala institucion - Cerca=0; Lejos=1 ==> "))
e= int(input("Ingresa su Situacion Laboral - Sin Trabajo=0; Con
Trabajo=1==> "))
f= int(input("Digite su Ingreso Familiar ==> "))
g= int(input("Ingresa su tipo de Vivienda(0=propia, 1=alquilada)==> "))

z=ModeloEnsamblado.predict([[a,b,c,d,e,f,g]])
if(z==0):
  print("El Estudiante tiene el ", accuracy_score(y_test, y_predC)*100,
"% de probabilidad de No Desertar")
else:
  print("El Estudiante tiene el ", accuracy_score(y_test, y_predC)*100,
"% de probabilidad de Desertar")
print("Los coeficientes del modelo son ", Modelo1.coef_)
print("El Intercepto ", Modelo1.intercept_)
```

*** Si bien es cierto que el modelo ensamblado ofrece un mismo porcentaje de confiabilidad que los modelos Regresión Logística Binaria y Naive Bayes, hay que resaltar que el modelo ensamblado ofrece mayor ROBUSTEZ que los anteriores. ***

ANEXO 3 Juicio de expertos

A continuación de la presente página.

CONSTANCIA DE VALIDACIÓN DE JUCIOS DE EXPERTOS

Yo, Yanny Milagritos Siguentes Díaz, con Documento Nacional de Identidad Nº 18090919, de profesión Ing. Computación y Sistemas grado académico Doctor, con número de colegiatura 54456, labor que ejerzo actualmente Docente Universidad Nacional de Trujillo.

Por medio de la presente hago constar que he revisado con fines de Validación el modelo predictivo propuesto en la tesis doctoral "**MODELO PREDICTIVO BASADO EN MACHINE LEARNING SUPERVISED Y LA DESERCIÓN ESTUDIANTIL EN CENTROS DE EDUCACIÓN SUPERIOR TECNOLÓGICOS PÚBLICOS DE LA REGIÓN LA LIBERTAD**".

Luego, de la revisión pertinente al modelo predictivo que propone el investigador, concluyo en las siguientes apreciaciones.

Criterios evaluados	Valoración positiva			Valoración negativa	
	MA	BA	DA	PA	D
El Modelo predictivo que se propone es creación propia del investigador.	X				
La tesis está dirigida a la deserción estudiantil de los IESTP de la región La Libertad, como referencia IESTP Trujillo.	X				
El modelo predictivo propuesto es auténtico, creativo e innovador.	X				

Apreciación total:
Muy de acuerdo (MA); Bastante de acuerdo (BA); De acuerdo (DA); Poco de acuerdo (PA); Desacuerdo (D)

Trujillo, a los 2 días del mes de mayo del 2024

Dr (a). Yanny Siguentes Díaz
DNI: 18090919

CONSTANCIA DE VALIDACIÓN DE JUCIOS DE EXPERTOS

Yo, ***Daniel Augusto Álvarez Campos***, con Documento Nacional de Identidad N° **44665640,** de profesión ***Ingeniero Informático***, grado académico Doctor, con número de colegiatura 127603, labor que ejerzo actualmente: ***Docencia Universitaria en Universidad Nacional de Trujillo.***

Por medio de la presente hago constar que he revisado con fines de Validación el modelo predictivo propuesto en la tesis doctoral "**MODELO PREDICTIVO BASADO EN MACHINE LEARNING SUPERVISED Y LA DESERCIÓN ESTUDIANTIL EN CENTROS DE EDUCACIÓN SUPERIOR TECNOLÓGICOS PÚBLICOS DE LA REGIÓN LA LIBERTAD**".

Luego, de la revisión pertinente al modelo predictivo que propone el investigador, concluyo en las siguientes apreciaciones.

Criterios evaluados	**Valoración positiva**			**Valoración negativa**	
	MA	**BA**	**DA**	**PA**	**D**
El Modelo predictivo que se propone es creación propia del investigador.	x				
La tesis está dirigida a la deserción estudiantil de los IESTP de la región La Libertad, como referencia IESTP Trujillo.	x				
El modelo predictivo propuesto es auténtico, creativo e innovador.	x				

Apreciación total:
Muy de acuerdo (MA); Bastante de acuerdo (BA); De acuerdo (DA); Poco de acuerdo (PA); Desacuerdo (D)

Trujillo, a los 2 días del mes de mayo del 2024

Dr. Daniel Augusto Álvarez Campos
DNI: 44665640.

CONSTANCIA DE VALIDACIÓN DE JUCIOS DE EXPERTOS

Yo, Lucía Margarita Saldaña Sáenz, con Documento Nacional de Identidad Nº 40699152, de profesión Docente, grado académico Doctora, con número de colegiatura 1540699152, labor que ejerzo actualmente Docencia Universitaria - UCV sede Lima Norte.

Por medio de la presente hago constar que he revisado con fines de Validación el modelo predictivo propuesto en la tesis doctoral "**MODELO PREDICTIVO BASADO EN MACHINE LEARNING SUPERVISED Y LA DESERCIÓN ESTUDIANTIL EN CENTROS DE EDUCACIÓN SUPERIOR TECNOLÓGICOS PÚBLICOS DE LA REGIÓN LA LIBERTAD**".

Luego, de la revisión pertinente al modelo predictivo que propone el investigador, concluyo en las siguientes apreciaciones.

Criterios evaluados	Valoración positiva			Valoración negativa	
	MA	BA	DA	PA	D
El Modelo predictivo que se propone es creación propia del investigador.	X				
La tesis está dirigida a la deserción estudiantil de los IESTP de la región La Libertad, como referencia IESTP Trujillo.	X				
El modelo predictivo propuesto es auténtico, creativo e innovador.	X				

Apreciación total:
Muy de acuerdo (MA); Bastante de acuerdo (BA); De acuerdo (DA); Poco de acuerdo (PA); Desacuerdo (D)

Trujillo, a los 2 días del mes de mayo del 2024

Dr (a). Lucía Margarita Saldaña Sáenz
DNI: 40699152