Facultad de Ingeniería

Escuela de Computación e Informática

SISTEMA CAUSAL PROBABILÍSTICO DE GESTIÓN EDUCATIVA

Predicción de deserción basado en explicación

Tesis de postgrado para optar al grado de Magíster en Ingeniería Informática

Autor:

Juan Ricardo Tarbes Vergara

Pamela Carolina Morales Vergara

Profesor guía:

PhD © Billy Mark Peralta Márquez

Santiago, Chile

2021

TABLA DE CONTENIDOS

INDICE DE TABLAS

INDICE DE FIGURAS

RESUMEN

En la actualidad, las instituciones de educación superior están tratando de contrarrestar los efectos de la reprobación en los cursos iniciales de las carreras, lo cual refleja una deficiencia en las aptitudes académicas en los alumnos, y que posteriormente puede conllevar a bajo desempeño a lo largo de la carrera. Estudios relacionados a esta problemática han sido realizados, utilizando variados métodos y analizando el contexto personal y social del estudiante. Mas son escasos los estudios que consideran la evaluación de aptitudes del estudiante como objetivo de análisis, y combinando con la metodología de modelos causales probabilísticos. Bajo el contexto anterior, se presenta una propuesta metodológica basada en la aplicación de un modelo causal probabilístico, como una herramienta de predicción oportuna para la toma de decisiones frente a posibles reprobaciones en el proceso de gestión educativa, utilizando como fuente de información la evaluación de aptitudes académicas. Para hacer frente a esta problemática se elige el uso de Redes Bayesianas, tomando una muestra de datos para estimar la probabilidad de las variables y sus dependencias, permitiendo identificar reglas interpretables que expliquen las razones por la que los alumnos reprueban, y así tomar medidas proactivas para evitar su ocurrencia. El presente trabajo, para el desarrollo del modelo elegido, se enfoca en los datos de un juego y una prueba diagnósticos realizados previamente a un curso de programación, y los resultados del posterior desempeño en este curso, que es transversal a los programas de estudio. La información obtenida corresponde al año 2019 de alumnos de primer año de carreras de Ingeniería de una universidad chilena.

ABSTRACT

Currently, higher education institutions are trying to counteract the effects of failure in the initial courses of careers, which reflects a deficiency in academic skills in students, and which can subsequently lead to poor performance throughout of the career. Studies related to this problem have been carried out, using various methods and analyzing the personal and social context of the student. But there are few studies that consider the evaluation of student aptitudes as an objective of analysis and combining it with the methodology of probabilistic causal models. Under the previous context, a methodological proposal based on the application of a probabilistic causal model is presented, as a timely prediction tool for decision-making in the face of possible failures in the educational management process, using as a source of information the evaluation of academic skills. To deal with this problem, the use of Bayesian Networks is chosen, taking a data sample to estimate the probability of the variables and their dependencies, allowing to identify interpretable rules that explain the reasons why students fail, and thus take proactive measures. to avoid its occurrence. The present work, for the development of the chosen model, focuses on the data of a game and a diagnostic test carried out prior to a programming course, and the results of the subsequent performance in this course, which is transversal to the study programs. The information obtained corresponds to the year 2019 of first-year engineering students from a Chilean university.

1 INTRODUCCIÓN

Estudios relacionados al rendimiento académico y deserción en universidades chilenas han sido realizados, los cuales han concluido que diversos factores como habilidades matemáticas y de lenguaje, satisfacción del estudiante, el género, si el estudiante estudia y trabaja (Barahona U, 2014), la prueba de selección universitaria (Aguirre, 2012) y las notas de enseñanza media (Vergara & Peredo, 2017) influyen significativamente en el rendimiento de los estudiantes, haciendo énfasis en el primer año de carrera.

Específicamente, en cuanto a la investigación de aptitudes intelectuales, el conjunto de universidades que han sido estudiadas es bastante amplio y heterogéneo en cuanto a sus características, estando ubicadas en distintos países tanto de América como de otros continentes, pero dado que es recomendable no generalizar en este tipo de estudio debido a que se considera de cierta complejidad (Garbanzo, 2007), y dado que actualmente no existe una gran cantidad de estudios sobre este aspecto en universidades chilenas, se crea la oportunidad de realizar una investigación que pudiera aportar nuevos conocimientos sobre el tema.

El presente trabajo de tesis tiene como principal objetivo identificar, aplicando modelos causales probabilísticos, las principales variables en el contexto de evaluación de aptitudes intelectuales, que influyen en el rendimiento académico de los alumnos de primer año de las carreras de ingeniería de una universidad chilena y que tienen como base común un curso de programación.

Para el estudio se utilizan, como fuente de información, los resultados de un juego diagnóstico realizado a los estudiantes antes de iniciar el curso de programación. El juego consta de seis niveles de dificultad, que permiten medir las habilidades lógicas que los alumnos poseen para encontrar, en el menor tiempo posible, soluciones óptimas a los problemas que le son presentados. Adicionalmente, se evalúan varias aptitudes en los estudiantes con una prueba diagnóstico, cuyos resultados también son utilizados como fuente de datos. Finalmente, las calificaciones del curso de programación completan la muestra de datos del estudio.

Actualmente existen estudios que predicen el abandono de los estudiantes como, por ejemplo, en un entorno de aprendizaje virtual (He, y otros, 2020) que utilizando información biográfica personal estadística y los datos de comportamientos secuenciales con VLE vía algoritmos de redes neuronales. Además, existe otro estudio que investiga las altas tasas de reprobación en estudiantes de cursos de introducción a la programación utilizando técnicas de minería de datos (Costa, Fonseca, Santana, Araújo, & Rego, 2017).

Los estudios presentados evalúan el abandono y la reprobación desde el análisis de datos, y lo que esta tesis investiga es una probabilidad más cercana a la realidad utilizando el modelo de Redes Bayesianas, basado en el Teorema de Thomas Bayes (1702-1761), clérigo del siglo XVIII que desarrolló una fórmula para el cálculo de probabilidades condicionales (Bayes, 1763).

$$P(A_i|B) = \frac{P(B|A_i)P(A_i)}{P(B)} = \frac{P(B|A_i)P(A_i)}{\sum_{j=1}^{n} P(B|A_j)P(A_j)}$$

Figura 1: Formula de Bayes, también conocida como Regla de Bayes

En esta investigación se plantean algunas preguntas, basadas en los objetivos del estudio, que pretenden ser respondidas a lo largo del documento, entre ellas, la suficiencia de los datos necesarios para realizar un buen estudio, o a como modelar causalmente la reprobación del curso interpretando los resultados en una Red Bayesiana e identificar los factores que desencadenan dicha la reprobación.

2 IDENTIFICACIÓN DEL PROBLEMA

Actualmente, en la Universidad Nacional Andrés Bello se está tratando de contrarrestar los efectos de la reprobación académica, los cuales ocasionan que alumnos en sus primeros años trunquen sus estudios sin posibilidad de retomarlos en un futuro cercano. En lo particular, en el primer año es dictado un curso de programación que es transversal a los programas de estudio de varias carreras de ingeniería, y dado que existe un porcentaje de estudiantes que reprueban este curso, por medio de la realización de pruebas diagnósticas previas, se busca mediante un modelo inferir sobre las causas de reprobación del curso.

2.1 Obtención de fuentes de datos

Para la realización del modelamiento causal, se utilizarán los resultados obtenidos de una prueba tipo juego y una prueba de diagnóstico realizadas a los estudiantes, con ambas se evalúan aptitudes del estudiante previamente al curso de programación, más los resultados de las evaluaciones parciales y finales del curso, siendo transversal a los programas de estudio. La muestra de datos es provista, por la Universidad Nacional Andrés Bello, por medio del profesor Pablo Hernan Schwarzenberg Riveros, quien es investigador en el área. La información obtenida corresponde al primer semestre del año 2019 de 467 alumnos de primer año de diversas carreras de ingeniería.

3 OBJETIVOS E HIPÓTESIS

3.1 Objetivo General

Realizar una predicción de reprobación, con el fin de identificar reglas interpretables que expliquen las razones por las cuales los alumnos reprueban y así tomar medidas proactivas que permitan evitar su ocurrencia, mediante el uso de Redes Bayesianas.

3.2 Objetivos Específicos

- Preparar el 100% de la base de datos para su utilización en el modelo Bayesiano en las fases iniciales del proyecto mediante métodos de limpieza y preprocesamiento.
- Identificar, como mínimo, un número de cinco variables relevantes que permitan realizar el estudio.
- Evaluar y seleccionar las herramientas de software que proporcionen las funciones requeridas para el modelamiento del problema mediante Redes Bayesianas.
- Modelar el problema de predicción usando una Red Bayesiana, que permita aplicar la utilización de las variables identificadas.
- Generar métricas para evaluar efectividad y desempeño del modelo generado.
- Evaluar los resultados obtenidos para identificar las causas que desencadenan la reprobación respondiendo a las preguntas que puedan ser planteadas.

3.3 Hipótesis

A partir de una muestra de datos obtenidos el año 2019, referente a evaluación de aptitudes para la programación de estudiantes universitarios de diversas carreras de ingeniería, es posible inferir, aplicando un modelo bayesiano, sobre las posibles causas de reprobación del curso de programación de primer año del plan de estudio.

3.4 Preguntas de investigación

- ¿Se cuenta con un muestreo de datos suficiente para realizar el estudio?
- ¿Existen datos duplicados o que no aportarán información al modelo y por lo tanto deben ser eliminados?
- ¿Qué factores desencadenan la reprobación del curso de programación?

- ¿Cuáles otros factores no representados en la fuente de datos pueden influir en la reprobación de los estudiantes?
- ¿Cómo interpretamos los resultados en una red bayesiana aplicada al problema de la desaprobación de un curso?
- ¿Como modelar causalmente la reprobación de un curso?

4 MARCO TEÓRICO

4.1 Rendimiento Académico asociado a la deserción

En un sentido muy amplio, la deserción universitaria se puede entender como renunciar a la universidad por razones personales; reprobar una o varias materias, ir a paso más lento o ser expulsado; cambiar de carrera y no recibirse (Rugarcía, 1993). En términos más generales, la reprobación de materias aparece como el aspecto de deserción más frecuente, dado ese escenario es que las universidades invierten recursos para lograr que estudiantes se nivelen y puedan sortear las complicaciones que implica enfrentarse al desafío de rendir en sus áreas académicas.

Son variados los factores que influyen en el rendimiento académico y varios los estudios sobre el tema, pero lo definen como el resultado cuantitativo obtenido durante el proceso de aprendizaje conforme a las evaluaciones que realiza el docente mediante pruebas objetivas y otras actividades complementarias (Saucedo, Herrera-Sánchez, Díaz, & Bautista, 2014).

Por tales razones se lleva a cabo esta investigación, basada en redes bayesianas, sobre cuáles variables del caso de estudio presentan las problemáticas que impiden que los estudiantes tengan un mejor rendimiento en el curso de programación de primer año de carrera.

4.2 Redes Bayesianas

(Neapolitan, 2003)

Las redes bayesianas son estructuras gráficas para representar las relaciones probabilísticas entre un gran número de variables y hacer inferencias probabilísticas con esas variables. En el libro *Learning Bayesian Networks* (Neapolitan, 2003) se detallan dos algoritmos para la inferencia exacta con variables discretas (algoritmo de transmisión de mensajes de Pearl) y para la inferencia probabilística simbólica (algoritmo de D’Ambrosio y Li).

Por otro lado, los diagramas de influencia nos dan una naturaleza gráfica de las redes bayesianas, entregando una compresión intuitiva de las relaciones entre las características.

El concepto de probabilidad nos otorga una mirada filosófica y nos muestra dos corrientes de interpretación, una frecuentista y otra con un grado de creencia. Donde, la primera considera la probabilidad como la frecuencia relativa de un experimento aleatorio y la segunda interpreta la probabilidad de manera subjetiva, y la utiliza para expresar su creencia respecto a una afirmación, dada cierta evidencia. En la frecuentista, la probabilidad obtenida no es propiedad de ninguno de los ensayos, sino que es una propiedad de toda la secuencia de ensayos y en la de grado de creencia asigna probabilidades a los eventos basada en la paridad de razones donde la relación (ratio) es 1/n. Esta última es la llamada "principio de indiferencia" (término popularizado por J.M. Keynes en 1921).

Para entender el concepto de probabilidad, este se define como un espacio de muestra $\Omega = \{e_1, e_2, \cdots e_n\}$ que es un conjunto y los resultados son los elementos del conjunto. Ejemplo:

$$\Omega = \{(1,1), (1,2), (1,3), (1,4), (1,5), (1,6), (2,1), (2,2), \cdots (6,5), (6,6)\}$$

En este espacio de muestra se encuentran las "variables aleatorias" que asigna un valor único a cada elemento (resultado) en el espacio de muestra. Se dice que una variable es "discreta" si su espacio es finito. Para el caso de un espacio de muestra finito, cada subconjunto del espacio de muestra se denomina "evento". Un subconjunto que contiene exactamente un elemento se denomina "evento elemental". Una vez que se identifica un espacio de muestra, una función de probabilidad se define de la siguiente manera:

$$P(E) = P\left(\{e_{i_1}\}\right) + P\left(\{e_{i_2}\}\right) + \cdots + P\left(\{e_{i_k}\}\right).$$

Donde el par (Ω, P) se denomina "espacio de probabilidad"

También, por su parte, tenemos el concepto de "probabilidad condicional"

$$P(E|F) = \frac{P(E \cap F)}{P(F)}$$

donde los eventos E y F son denotados como $P(E|F)$, E dado F, que significa que la probabilidad de que ocurra E dado que sabemos que F ha ocurrido. A su vez dos eventos E y F son "condicionalmente independientes" cuando la probabilidad de cada uno de ellos no está influenciada por que el otro evento ocurra o no, es decir, cuando ambos eventos no están relacionados.

Con el Teorema de Bayes se pueden calcular las probabilidades condicionales de eventos de interés a partir de probabilidades conocidas.

$$P(E|F) = \frac{P(F|E)P(E)}{P(F)}$$

Además, dados n eventos mutuamente excluyentes y exhaustivos $E_1, E_2, \ldots E_n$ tal que $P(E_i) \neq 0$ para todos los i, tenemos que $1 \leq i \leq n$,

$$P(E_i|F) = \frac{P(F|E_i)P(E_i)}{P(F|E_1)P(E_1) + P(F|E_2)P(E_2) + \cdots P(F|E_n)P(E_n)}$$

Para calcular la probabilidad condicional utilizando cualquiera de las dos fórmulas se denomina "inferencia bayesiana".

La inferencia bayesiana es una técnica estadística que permite mediante una distribución de probabilidad, ajustar el modelo probabilístico, permitiendo obtener información de los parámetros sobre los cuales se desea realizar alguna estimación.

En base a la definición de "variable aleatoria", que indica que, una variable aleatoria X representa a cualquiera de un conjunto de valores del llamado espacio de X, se tiene una definición directa de una "distribución de la probabilidad conjunta" que se refiere a cuando dos variables aleatorias inducen a una función de probabilidad en el producto cartesiano de sus espacios. Ejemplo:

$P\ (x, y)$ que es la **distribución de probabilidad conjunta** de X e Y

$$\sum_{x_1, x_2, \cdots x_n} P(X_1 = x_{1,}\, X_2 = x_{2,} \cdots X_n = x_n) = 1$$

Gráficamente, en una red bayesiana, cada nodo representa a una variable y cada arco que une los nodos indican relaciones de influencia causal. Una red bayesiana se compone de variables dependientes e independientes, donde, las independientes corresponden a los nodos padres y las dependientes a los nodos hijos.

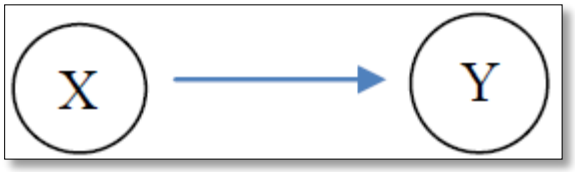

Figura 2: Ejemplo de una relación de influencia causal

El nodo con la variable X es padre del nodo de la variable Y (hijo).

La independencia condicional se distingue en tres tipos de nodos de acuerdo con las direcciones de los arcos que inciden en el nodo:

- Nodos en secuencia: $X \rightarrow Y \rightarrow Z$.
- Nodos divergentes: $X \leftarrow Y \rightarrow Z$.
- Nodos convergentes: $X \rightarrow Y \leftarrow Z$.

La topología o estructura de la red nos da información sobre las dependencias probabilísticas entre las variables. La red también representa las independencias condicionales de una variable (o conjunto de variables) dada(s) otra(s) variable(s). Por ejemplo, la Figura 3 ***reacciones*** es condicional e independiente. de C, G, F, D, dado ***tifoidea***. (Donde: C es ***comida***, T es ***tifoidea***, G es ***gripe***, R es ***reacciones***, F es ***fiebre*** y D es ***dolor***) (Sucar, 2006).

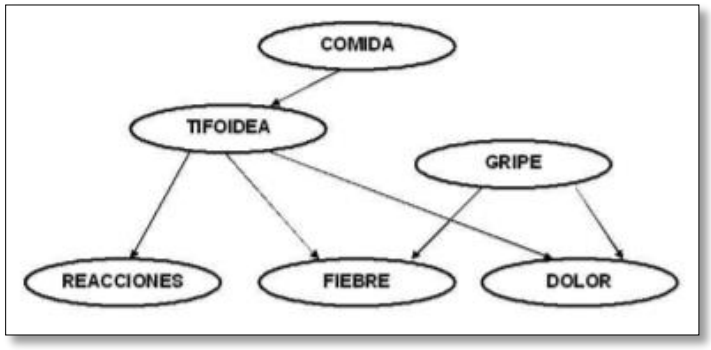

Figura 3: Ejemplo de una Red Bayesiana

Las redes bayesianas representan explícitamente nuestro conocimiento sobre los elementos en el sistema y las relaciones que existen entre ellos. De esta manera, se pueden “aprender” las probabilidades de todos los elementos de la red a partir del conocimiento de algunos de ellos y de las relaciones condicionales entre ellos (López Balanzátegui, Flores Herrera, Flores Nicolalde, & Flores Nicolalde, 2016).

4.3 Investigación sobre estudios relacionados a la predicción académica

Se realizó una investigación de varios estudios relacionados al uso de Redes Bayesianas aplicados a las predicciones de rendimiento académico, encontrando cinco muy relevantes para nuestra investigación:

4.3.1 Análisis de datos de estudiantes de ingeniería para la predicción del rendimiento académico mediante técnica de clasificación bayesiana

(Sánchez Guzmán & Rico Páez, 2018)

En este trabajo se presenta un estudio para analizar datos de estudiantes de ingeniería para desarrollar un modelo predictivo del rendimiento académico por medio de la técnica de clasificación bayesiana (Bayes Ingenuo) con el objetivo de predecir el rendimiento académico e identificar los principales factores que inciden en éste.

En el desarrollo al descubrimiento del conocimiento en la base de datos se usó la técnica de minería de datos KDD, donde se realizó la “recolección de datos” correspondientes a 306 estudiantes de 7 cursos de primer y segundo semestre de ingeniería. La información de aprobación y reprobación fue proporcionada por los docentes de la institución y el resto de las variables fueron recopiladas por medio de encuestas. Luego se realizó el “preprocesado” de datos” que transformó la información de tal manera que puedan ser usados por la técnica de minería de datos a usar. Se contó con 21 atributos, donde el atributo “apruebo” se definió con la etiqueta de la clase. En el proceso de preprocesado se descubre que los datos están desbalanceados y se procede a la utilización del método de ganancia de información. Posteriormente, se procede a evaluar el modelo predictivo por medio de exactitud de las predicciones utilizando el método de validación cruzada

que divide aleatoriamente los datos de entramiento, uno para predecir los resultados y el segundo para calcular su exactitud.

Al usar los 12 mejores atributos mejor clasificados se construyó un software predictor en el que se programó el algoritmo de Bayes Ingenuo y se disponibilizó en formato HTML5 con el objetivo de ser publicado en un sitio web

4.3.2 Predicción del rendimiento académico de los estudiantes de física a través de las redes bayesianas en la unidad de cantidad de movimiento lineal

(López Balanzátegui, Flores Herrera, Flores Nicolalde, & Flores Nicolalde, 2016)

Este trabajo presenta una investigación de probabilidades estadísticas con Redes Bayesianas, basándose en los resultados de las evaluaciones realizadas a estudiantes de física de una universidad ecuatoriana, con lo cual, se pudo inferir resultados futuros del desempeño de los estudiantes y la relación de los conocimientos teóricos con la resolución de problemas en la materia de física.

En el documento se hace una descripción conceptual de las redes bayesianas y teoría de grafos. Además, de una descripción de conceptos de física como el movimiento lineal.

El método utilizado se centra en el área temática referente a la enseñanza de la física, estadística e investigación. Este se desarrolla a través de una metodología cualitativa y cuantitativa. La metodología es correlacional, es decir en la relación entre los resultados de aprendizaje basados en pruebas, y su efecto en el rendimiento académico de los estudiantes

En el estudio participaron 27 estudiantes que cursan primer año de ingenierías informáticas en la materia de física en donde se avaluó mediante pruebas conceptuales y resolución de problemas la unidad de movimiento lineal.

El proceso de análisis de datos fue realizado en una hoja de Excel para poder inferir en forma futura el resultado de las evaluaciones, relacionando estos con la probabilidad de que el estudiante aprueba o no la materia. En cierta medida se realizó una discretización de los datos.

Mediante la utilización del software ELVIRA se diseñaron 8 redes bayesianas que calcularon las probabilidades a priori y posteriori los resultados obtenidos demuestran que en la materia de física es muy importante conocer la teoría para poder aplicarla a la resolución de problemas.

4.3.3 Análisis de datos educativos utilizando Redes Bayesianas

(Oviedo, Puris, Villacís, Delgado, & Moreno, 2015)

Este trabajo postula que una red bayesiana es un proceso de aprendizaje que se divide en dos etapas: el aprendizaje estructural y el aprendizaje paramétrico. Donde la primera, consiste en obtener la estructura de red bayesiana (sus relaciones dependencias e independencias) y la segunda obtener las probabilidades a priori y condicionales requeridas a partir de una estructura dada. El trabajo hace uso de modelos probabilísticos gráficos para determinar el problema de la deserción estudiantil.

La base de conocimiento es de 733 estudiantes matriculados en el periodo 2012-2013 de la universidad Técnica Estatal de Quevedo.

El documento hace una descripción de las redes bayesianas, aprendizaje bayesiano y un estudio de los parámetros propuestos realizando un análisis experimental. En la experimentación de los datos el software ELVIRA fue utilizado, identificando 17 variables con las que se trabajó y se discretizó por medio de un proceso manual a través de criterio de expertos. Las variables clasificadas fueron usadas para construir la red bayesiana a partir de un clasificador de Bayes Ingenuo. Los resultados evidencian que un factor de impacto para la deserción estudiantil es el año en el que está cursando y los diferentes factores socio económicos que de manera directa influyen en el desempeño académico del estudiante.

4.3.4 Análisis de la progresión de los estudiantes en una asignatura introductoria a la programación mediante Redes Bayesianas

(Marco Galindo, Minguillón, & Sancho-Vinuesa, 2020)

Este trabajo busca encontrar una relación entre el perfil de los estudiantes y su grado de participación en la asignatura introductoria de programación, para ello se proponen 8 actividades optativas como pruebas de evaluación continua que se combinan con la realización de un ejercicio práctico de programación más complejo y obligatorio que integra todos los contenidos del curso. Para el estudio se utilizaron 1043 registros académicos de 3 semestres consecutivos (2017-2; 2018-1; 2018-2) donde comparan diferentes métodos de clasificación de inferencia bayesiana usando los resultados obtenidos por los estudiantes en los primeros ejercicios planteados.

El objetivo es relacionar las entregas que hacen los estudiantes con el resultado obtenido en la actividad obligatoria, para lo cual se construyeron dos modelos diferentes para establecer la relación entre las variables. Para la creación de la red bayesiana se usó el paquete de R “bnlearn”.

Los resultados permiten observar que conforme aumenta el número de actividades consecutivas no entregadas, la probabilidad de superar la actividad obligatoria disminuye.

4.3.5 Análisis del alumnado de la Universidad de Almería mediante Redes Bayesianas

(Morales & Salmerón, 2003)

Este trabajo hace uso de las redes bayesianas para la clasificación o la extracción de perfiles. El objetivo es construir una red bayesiana que modelice a los alumnos matriculados en la Universidad de Almería en el curso 2000/2001 y poder realizar inferencias sobre dicha red.

Para el estudio se dispuso de una base de datos con 13.747 alumnos donde se tomaron 22 variables que fueron adaptados para ser usados con el software Elvira (Entorno de Desarrollo para Modelos Gráficos Probabilísticos) que consta de 3 modos básicos:

Edición, Inferencia y Aprendizaje. Con este último se construye la tabla de probabilidad y la estructura de la red bayesiana haciendo uso del algoritmo Spirtes, Glymour, and Scheines (1993). A continuación, se hace un análisis de variables mediante propagación de probabilidades para la obtención de los perfiles buscados.

4.4 Metodologías de ciencia de datos

La presente investigación no se enfoca en el estudio de las metodologías para ciencia de datos existentes, claro está, son mencionadas como parte del estudio.

4.4.1 Proceso KDD

El término KDD significa “Descubrimiento de Conocimiento en Base de Datos (Knowledge Discovery in Databases)” e implica la evaluación e interpretación de patrones y modelos para tomar decisiones con respecto a lo que constituye conocimiento y lo que no lo es (KDD Knowledge Discovery in Databases).

El proceso KDD se puede definir como “el proceso no trivial de identificar patrones válidos, novedosos, potencialmente útiles y, en última instancia, comprensibles en los datos” (Fayyad & Piatetsky-Shapiro, 1996). KDD se centra en el proceso general de descubrimiento de conocimientos a partir de los datos, incluido cómo se almacenan y se accede a los datos, cómo se pueden escalar los algoritmos a conjuntos de datos masivos y seguir funcionando de manera eficiente, cómo se pueden interpretar y visualizar los resultados, y cómo la interacción general entre humanos y máquinas se puede modelar y apoyar. KDD pone especial énfasis en encontrar patrones comprensibles que puedan interpretarse como conocimientos útiles o interesantes (Fayyad & Piatetsky-Shapiro, 1996).

Como se aprecia en la Figura 4, el proceso KDD es iterativo ya que la salida de cada fase puede retroceder a los pasos anteriores y porque, a veces, son necesarias varias iteraciones

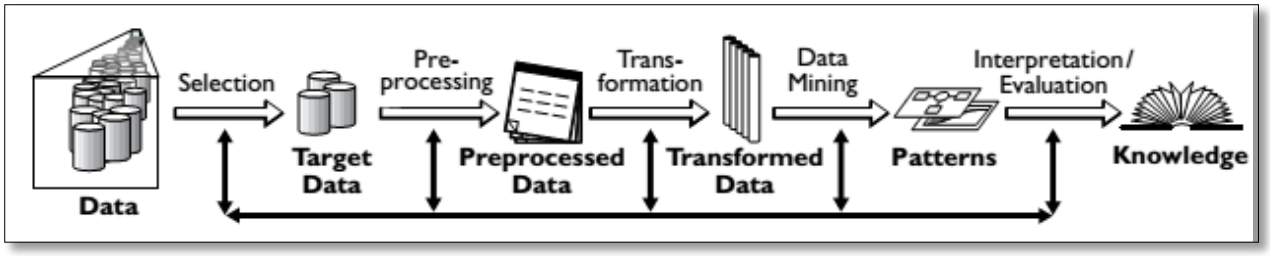

Figura 4: Proceso KDD

El proceso es interactivo e iterativo que involucra numerosos pasos, resumidos como:

i. Aprender el dominio de la aplicación: Incluye conocimientos previos relevantes y los objetivos de la aplicación.
ii. Crear un conjunto de datos de destino: Incluye seleccionar un conjunto de datos o centrarse en un subconjunto de variables o muestras de datos en las que se realizará el descubrimiento.
iii. Limpieza y preprocesamiento de datos: Incluye operaciones básicas, como eliminar el ruido o valores atípicos si corresponde, recopilar la información necesaria para modelar o contabilizar el ruido, decidir estrategias para manejar los campos de datos faltantes y contabilizar la información de secuencia de tiempo y los cambios conocidos, así como decidir problemas de DBMS, como tipos de datos, esquema y mapeo de valores perdidos y desconocidos.
iv. Reducción y proyección de datos: Incluye encontrar características útiles para representar los datos, dependiendo del objetivo de la tarea, y usar métodos de reducción o transformación de dimensionalidad para reducir el número efectivo de variables bajo consideración o encontrar representaciones invariantes para los datos.
v. Elegir la función de la minería de datos: Incluye decidir el propósito del modelo derivado del algoritmo de minería de datos. (por ejemplo, resumen, clasificación, regresión y agrupamiento).
vi. Elección de los algoritmos de minería de datos: Incluye la selección de métodos que se utilizarán para buscar patrones en los datos, como decidir qué modelos y parámetros pueden ser apropiados (por ejemplo, los modelos para datos categóricos son diferentes de los modelos en vectores sobre reales) y hacer

coincidir un método de minería de datos en particular con los criterios generales del proceso KDD (por ejemplo, el usuario puede estar más interesado en comprender el modelo que en sus capacidades predictivas).

vii. Minería de datos: Incluye la búsqueda de patrones de interés en una forma de representación particular o un conjunto de tales representaciones, incluidas reglas o árboles de clasificación, regresión, agrupación, modelado de secuencias, dependencia y análisis de líneas.

viii. Interpretación: Incluye interpretar los patrones descubiertos y posiblemente volver a cualquiera de los pasos anteriores, así como la posible visualización de los patrones extraídos, eliminar patrones redundantes o irrelevantes y traducir los útiles en términos comprensibles para los usuarios.

ix. Usar el conocimiento descubierto: Incluye incorporar este conocimiento en el sistema de desempeño, tomar acciones basadas en el conocimiento o simplemente documentarlo y reportarlo a las partes interesadas, así como verificar y resolver posibles conflictos con conocimiento previamente creído (o extraído).

(Fayyad & Piatetsky-Shapiro, 1996)

4.4.2 Metodología CRISP-DM

CRISP-DM (Cross Industry Standard Process for Data Mining) proporciona una descripción normalizada del ciclo de vida de un proyecto estándar de análisis de datos, de forma análoga a como se hace en la ingeniería del software con los modelos de ciclo de vida de desarrollo de software

La metodología contempla seis fases las que son:

i. Comprensión del Negocio: Se enfoca en la comprensión de los objetivos del proyecto, las necesidades del cliente.

ii. Entendimiento de los datos: Esta fase comienza con la colección de datos inicial y continua con las actividades que permiten familiarizarse con los datos, identificar los problemas de calidad, descubrir conocimiento preliminar sobre los datos, y/o descubrir subconjuntos interesantes para formar hipótesis en cuanto a la información oculta.

iii. Preparación de datos: La fase de preparación de datos cubre todas las actividades necesarias para construir el conjunto final de datos (los datos que se utilizarán en las herramientas de modelado) a partir de los datos en bruto iniciales. Las tareas incluyen la selección de tablas, registros y atributos, así como la transformación y la limpieza de datos para las herramientas que modelan.

iv. Modelado: En esta fase, se seleccionan y aplican las técnicas de modelado que sean pertinentes al problema (cuantas más mejor), y se calibran sus parámetros a valores óptimos. Típicamente hay varias técnicas para el mismo tipo de problema de minería de datos. Algunas técnicas tienen requerimientos específicos sobre la forma de los datos. Por lo tanto, casi siempre en cualquier proyecto se acaba volviendo a la fase de preparación de datos.

v. Evaluación: En esta etapa en el proyecto, se han construido uno o varios modelos que parecen alcanzar calidad suficiente desde la una perspectiva de análisis de datos. Antes de proceder al despliegue final del modelo, es importante evaluarlo a fondo y revisar los pasos ejecutados para crearlo, comparar el modelo obtenido con los objetivos de negocio. Un objetivo clave es determinar si hay alguna cuestión importante de negocio que no haya sido considerada suficientemente. Al final de esta fase, se debería obtener una decisión sobre la aplicación de los resultados del proceso de análisis de datos.

vi. Despliegue: Generalmente, la creación del modelo no es el final del proyecto. Incluso si el objetivo del modelo es de aumentar el conocimiento de los datos, el conocimiento obtenido tendrá que organizarse y presentarse para que el cliente pueda usarlo. Dependiendo de los requisitos, la fase de desarrollo puede ser tan simple como la generación de un informe o tan compleja como la realización periódica y quizás automatizada de un proceso de análisis de datos en la organización.

(Román, 2016)

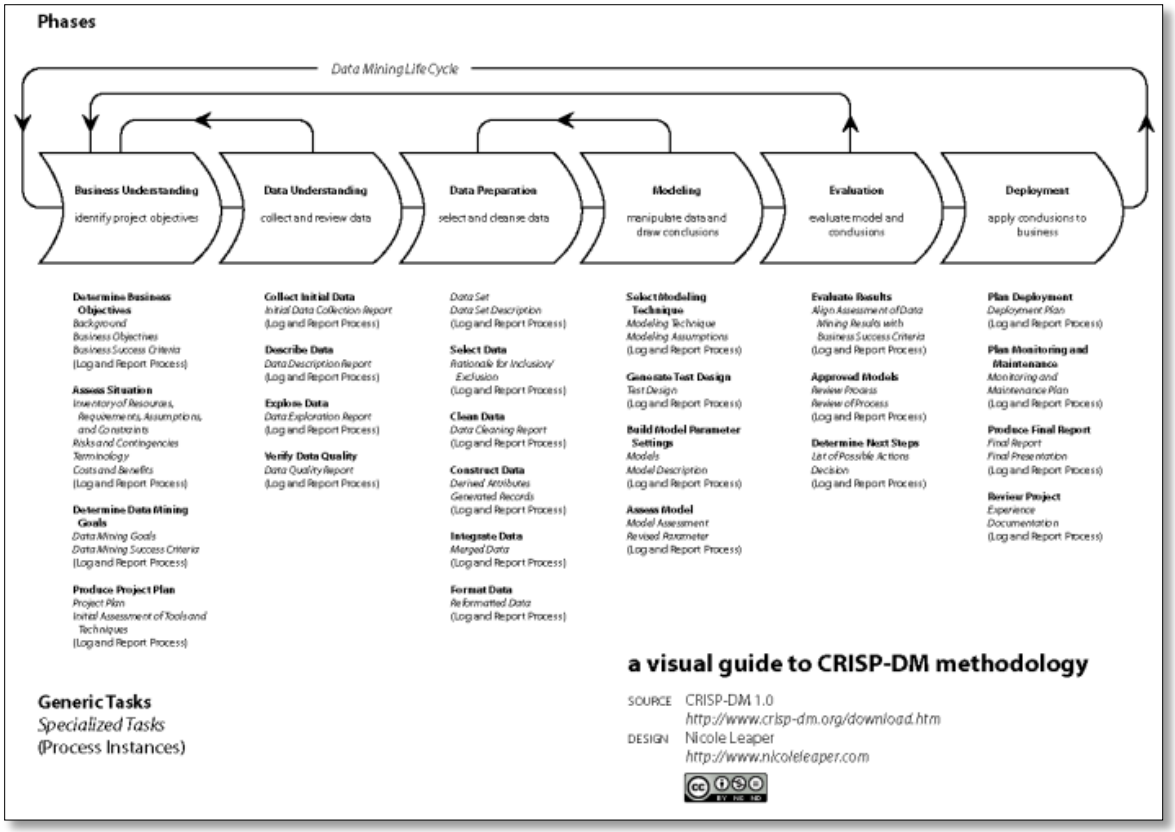

Figura 5: Metodología CRISP-DM

Esta metodología cuenta, por cada fase con un conjunto de tareas y actividades asociadas a cada tarea.

4.4.3 Metodología SEMMA

La metodología SEMMA, creada por el SAS Institute, se define como el proceso de selección, exploración y modelado de grandes cantidades de datos para revelar patrones de negocio desconocidos (Institute, 1998). El nombre de esta terminología es el acrónimo correspondiente a las cinco fases básicas del proceso: Sample (Muestreo), Explore (Exploración), Modify (Modificación), Model (Modelado), Assess (Valoración).

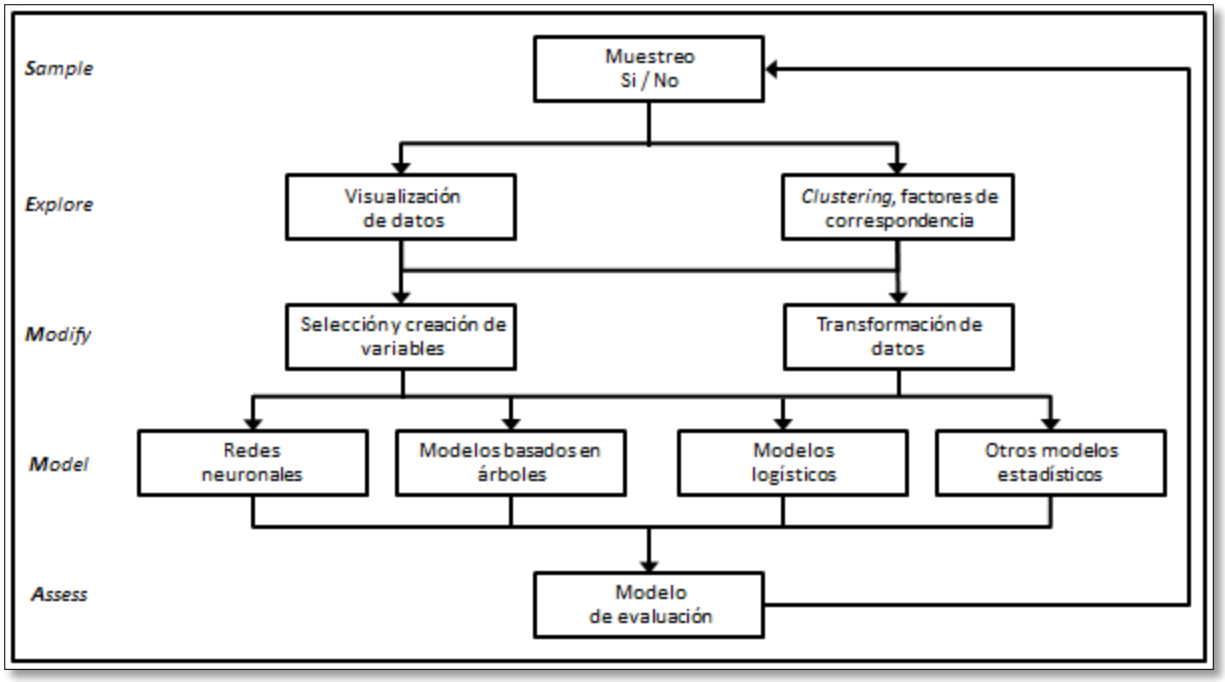

Figura 6: Metodología SEMMA

En la figura 6 se visualiza la dinámica general de la metodología que describe las fases de esta:

i. Muestreo: En esta primera fase de la metodología, se realiza la extracción de un conjunto de datos (población muestral) sobre la que se va a llevar a cabo el análisis.

ii. Exploración: En esta fase, se realiza un análisis de los datos extraídos en la muestra, para lo cual se propone el uso de herramientas de visualización o de diferentes técnicas estadísticas para la exploración de la información seleccionada, que contribuyan a poner de manifiesto relaciones entre variables.

iii. Modificación: La tercera fase de la metodología, involucra la modificación de los datos que van a ser ingresados al modelo para que tengan el formato adecuado, mejorando la definición de estos.

iv. Modelado: En esta fase, se procede a modelar el conjunto de datos, permitiendo al software realizar una búsqueda completa de combinaciones de datos que ayudarán a predecir los resultados esperados de manera confiable.

v. Valoración: La última fase de la metodología SEMMA, consiste en la valoración de los datos obtenidos para determinar el grado de confiabilidad de estos y así poder evaluar el modelo, mediante la comparación con otros métodos estadísticos o con nuevas poblaciones muestrales.

(Peralta, 2014)

4.5 Algoritmos y Técnicas para la minería de datos

La minería de datos se presenta como una tecnología de apoyo para explotar, analizar, comprender y aplicar el conocimiento obtenido usando grandes volúmenes de datos y descubrir patrones que ayuden a la identificación de estructuras de datos (Rodríguez Suárez & Díaz Amador, 2009).

Las técnicas y algoritmos de minería de datos para la búsqueda de patrones y la extracción de información que se ocultan en grandes cantidades de información se tienen:

- **Algoritmos supervisados o predictivos**: Predicen el valor de un atributo de un conjunto de datos, conocidos otros atributos.
- **Algoritmos no supervisados o del descubrimiento del conocimiento**: Estos algoritmos descubren patrones y tendencias en los datos actuales.
- **Técnica de almacenamiento de datos**: Es un proceso de organización de grandes cantidades de datos de diversos tipos guardados en una organización con el objetivo de facilitar su recuperación con fines analíticos.
- **Técnica de análisis exploratorio de datos (EDA)**: Tiene como objetivo determinar las relaciones entre las variables cuando no hay o no está totalmente definida la naturaleza de estas relaciones. Abarcan desde los métodos estadísticos simples a los más avanzados como las técnicas de exploración de multivariables.
- **Técnica de redes neuronales**: Son técnicas analíticas que permiten modelar el proceso de aprendizaje de una forma similar al funcionamiento del cerebro humano, es decir, aprender a partir de nuevas experiencias.

Esta última técnica tiene la característica de trabajar con datos incompletos. Además, pose dos formas de aprendizaje: supervisado y no supervisado. Entre estas podemos encontrar:

- **Análisis Preliminar de datos usando Query Tools**: Aplicación de consultas SQL al conjunto de datos.
- **Técnicas de visualización**: Ubicación de patrones, se usa al comienzo de un proceso de minería de datos para determinar la calidad de los datos.
- **Reglas de Asociación**: Establece asociaciones en base a los perfiles de los clientes sobre los cuales se realiza la minería de datos.
- **Algoritmos Genéticos**: Son técnicas de optimización que usan procesos tales como la combinación genética y mutaciones.
- **Redes Bayesianas**: Determinan relaciones causales que expliquen un fenómeno según los datos contenidos en una base de datos. Se han usado principalmente para realizar predicciones.
- **Árbol de Decisión**: Estructuras que representan conjuntos de decisiones, y estas decisiones generan reglas para la clasificación de un conjunto de datos.
- **Clustering (agrupamiento)**: Agrupan datos dentro de un número de clases preestablecidas o no.
- **Segmentación**: Consiste en la división de la totalidad de los datos, según determinados criterios.
- **Clasificación**: Consiste en definir una serie de clases, donde poder agrupar los diferentes clientes.
- **Predicción**: Consiste en intentar conocer resultados futuros a partir de modelizar los datos actuales.

El usar minería de datos contribuye a la toma de decisiones tácticas y estratégicas para abrir nuevas oportunidades de negocio, proporciona poder de decisión a los usuarios y es capaz de medir las acciones y resultados de una mejor manera.

4.6 Metodología de gestión de proyectos

En la actualidad existen dos enfoques para la gestión de proyectos que corresponden al estándar del PMI que se basa en PMBOK y el segundo enfoque está relacionado a las metodologías ágiles, tal como SCRUM.

4.6.1 PMBOK

PMBOK es una guía desarrollada por el Project Managment Institute (PMI) y entrega las mejores prácticas relacionadas a la gestión, administración y la dirección de proyectos mediante técnicas y herramientas. Esta guía abarca 5 macroprocesos, 10 áreas de conocimiento y 49 procesos que se pueden ver en el siguiente cuadro resumen:

Tabla 1: Relación entre las áreas de conocimiento y los macroprocesos

4.6.2 SCRUM

La metodología Scrum es un marco de trabajo o framework que es utilizado para trabajar colaborativamente en equipo y obtener el mejor resultado posible de un proyecto. Se trata de una metodología de trabajo ágil que tiene como finalidad entregar valor en periodos cortos de tiempo. Scrum se basa en aspectos tales como la flexibilidad en la adopción de cambios, el factor humano, la colaboración e interacción con el cliente, el desarrollo iterativo como forma de asegurar buenos resultados.

En Scrum un proyecto se ejecuta en ciclos temporales cortos y de duración fija, donde cada iteración tiene que proporcionar un resultado completo, un incremento del producto

final. Según el sitio proyectosagiles.org el proceso de scrum se puede ver reflejado en la siguiente imagen.

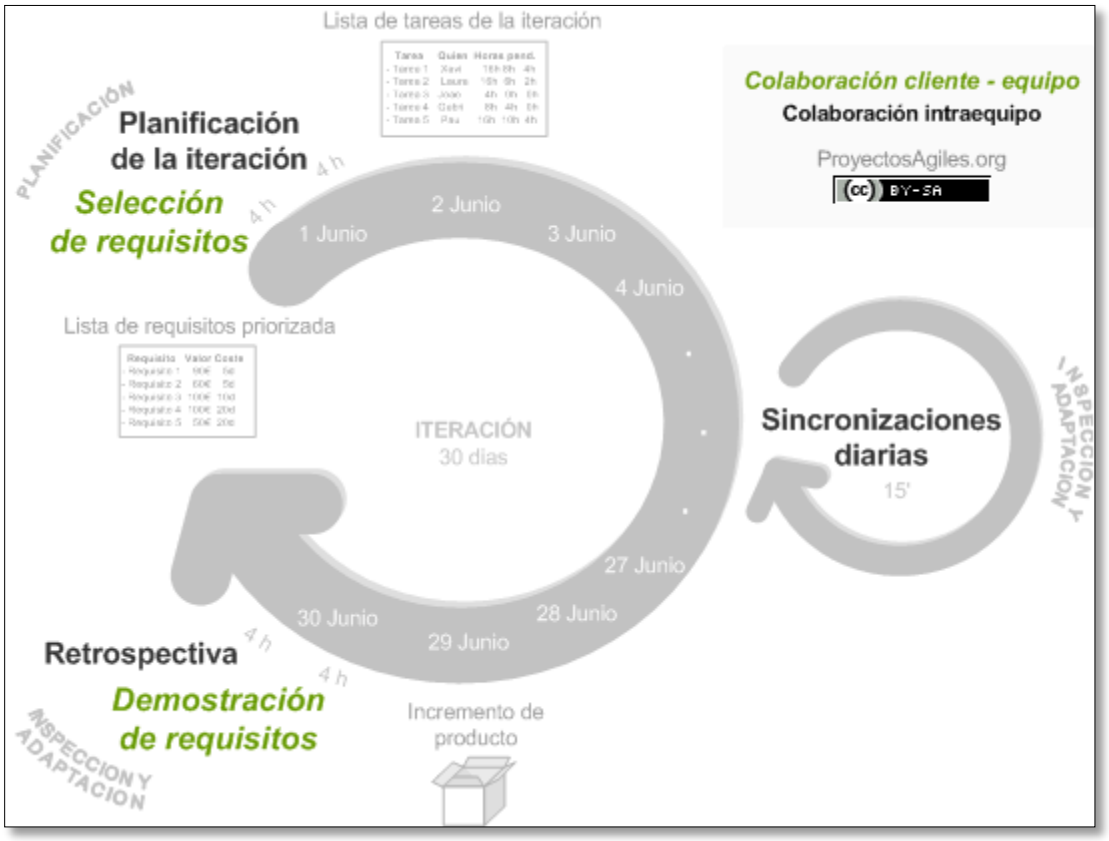

Figura 7: Proceso de Scrum

5 METODOLOGÍA DE TRABAJO

5.1 Enfoque de la investigación

Dado el planteamiento del problema que se desea resolver, la naturaleza de la información a utilizar llevado a los resultados que se quieren obtener y concluir, este estudio entregará un análisis, por medio del uso de Redes Bayesianas, de una predicción de aprobación / reprobación basada en explicación de un curso transversal de carreras de ingeniería.

Se utilizará un enfoque cuantitativo, por medio de la observación analítica, que permitirán determinar las variables que sustenten los resultados obtenidos.

El tipo de investigación a seguir es explicativo, dado que la finalidad es hallar las razones por los cuales ocurren los hechos.

5.2 Metodología de análisis de datos

Dentro del estudio se analizaron tres marcos metodológicos sobre el desarrollo de un trabajo de ciencia de datos, estos comparten ciertas similitudes tal y como lo plantea la publicación sobre el estudio comparativo de estas tres metodologías (Azevedo & Santos, 2008) y que sirve como base para la toma de decisión del método a seguir. La siguiente tabla es un cuadro comparativo de estas tres metodologías.

KDD	SEMMA	CRISP-DM
Pre KDD	----------	Comprensión del negocio
Selección	Muestreo	Compresión de los datos
Preprocesamiento	Exploración	
Transformación	Modificación	Preparación de los datos
Minería de datos	Valoración	Modelado
Interpretación y evaluación	Evaluación	Evaluación
Post KDD	----------	Despliegue

Tabla 2: Resumen comparativo entre KDD, SEMMA y CRISP-DM (Azevedo & Santos, 2008)

Según el sitio KDnuggets, CRISP-DM sigue siendo la metodología más popular para proyectos de análisis, minería de datos y ciencia de datos. Dicha afirmación se basa en la encuesta actualizada que realizó (KDnuggets, 2014) donde se responde la pregunta

¿Qué metodología principal está utilizando para sus proyectos de análisis, minería de datos o ciencia de datos? Los resultados de esta encuesta se pueden ver en el siguiente gráfico.

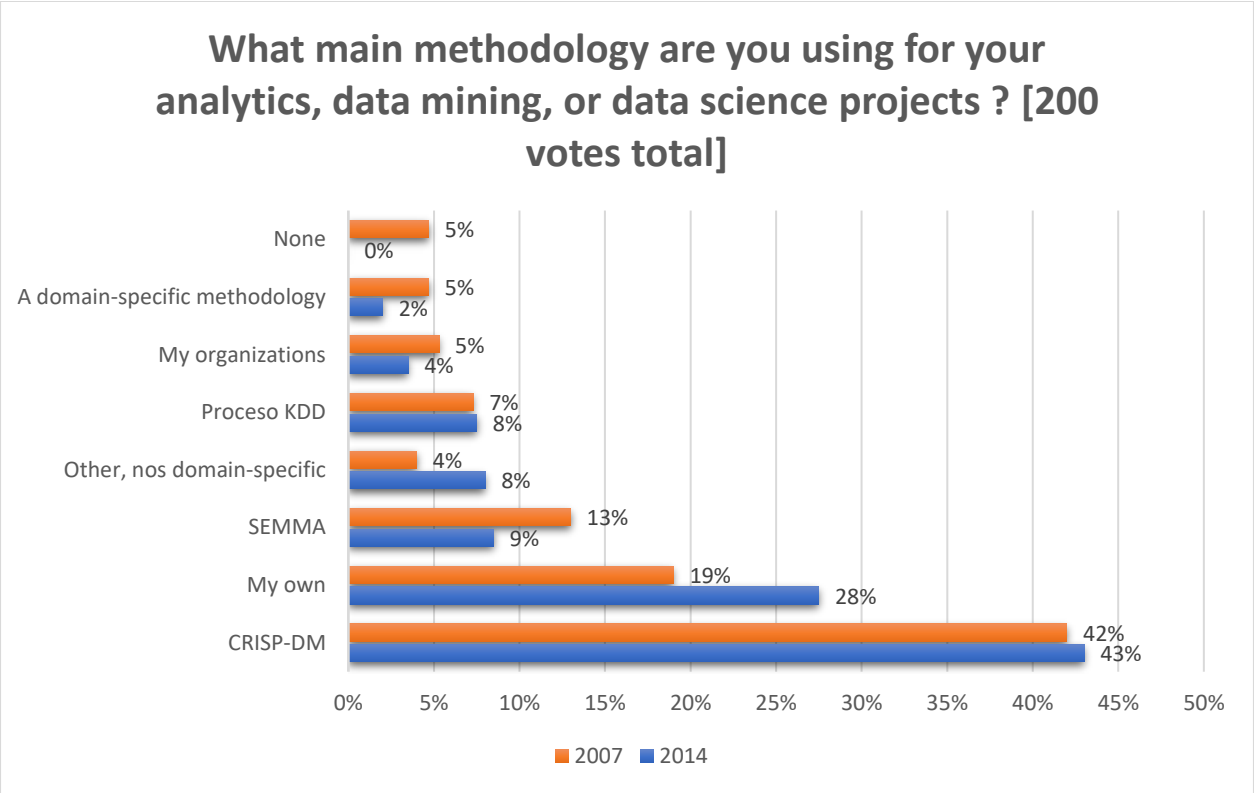

Figura 8: Encuesta KDnuggets sobre métodos análisis, minería de datos o ciencia de datos

En base a la información presentada, y tomando como referencia la conclusión de la publicación (Azevedo & Santos, 2008), tanto SEMMA como CRISP-DM son implementaciones del proceso KDD descrito por (Fayyad & Piatetsky-Shapiro, 1996), por tal razón, y dada la naturaleza del análisis que se desea realizar para este estudio, se considera que el uso del proceso KDD es el más adecuado dado que **proporciona una guía general** del trabajo a realizar en cada fase y a partir de él se puede realizar una variación del método para ser ajustado al proceso a utilizar en este estudio, que es más acotado en cuanto a su implementación completa, considerando las siguientes fases:

- Limpieza y preprocesamiento de datos
- Reducción y proyección de datos
- Elección de los algoritmos de minería de datos

- Minería de datos
- Interpretación
- Usar el conocimiento descubierto

No se consideran las fases iniciales del proceso ya que este estudio parte de la base de una información seleccionada y que debe ser analizada.

5.3 Obtención de datos

Los datos son proporcionados por la Universidad Andrés Bello y corresponden al primer semestre del año 2019 de alumnos de primer año de diversas carreras de ingeniería. El universo por utilizar en este estudio consta de 467 registros de alumnos que realizaron la prueba inicial.

5.4 Metodología de gestión de proyectos

El proyecto se enfoca en la utilización del estándar del Project Management Institute (PMI) para la administración de proyectos versión 6 que contiene un enfoque agile en relación con su versión anterior. Actualmente, está próxima la versión 7 que contendrá modificaciones importantes, como lo es un enfoque no en procesos, sino que en principios

Tanto el estándar del PMI como la metodología ágil para la gestión y administración de proyectos presentan ventajas y desventajas entre ellas, pero el proyecto de análisis que se realiza en este estudio se inclina por el estándar del PMI ya que este presenta una ventaja relacionada al conocimiento de los requerimientos que, en este caso, son concretos dada la naturaleza del proyecto y no variaran en el tiempo permitiendo llevar un control secuencial del estado del proyecto en el tiempo.

5.4.1 Hitos del proyecto

Los hitos son aquellas tareas que permiten controlar el avance del proyecto, estos van marcando los logros objetivos del proyecto para ayudar a determinar posibles desviaciones cuando se realiza el seguimiento y control de este.

Los siguientes son los hitos que se determinaron para el proyecto.

Hito	Punto de control	Descripción
1	Inicio del proyecto	Indica el comienzo oficial del trabajo, el kickoff donde se presenta el proyecto y los interesados.
2	Alcance del proyecto	Permite determinar que ya se ha definido todo el trabajo a realizar para dar cumplimiento de los objetivos planteados.
3	Marco Teórico	Marca el cumplimiento de todo el trabajo investigativo en relación con la búsqueda de información que de soporte al trabajo a realizar.
4	Metodología de trabajo	Indicará la culminación de la definición de los métodos a utilizar para analizar el tema a tratar, la estrategia que vincula todas las etapas de la investigación.
5	Preprocesamiento de datos	Este punto de control indica que se logró un entendimiento acabado de la información base utilizada para la investigación.
6	Implementación de algoritmos	Hito crucial que indica la utilización de los algoritmos seleccionados para llevar a cabo los planteamientos indicados en el alcance del proyecto.
7	Resultados y conclusiones	Este hito marca la finalización del trabajo de investigación, permitiendo entregar resultados concretos y fundamentados.

Tabla 3: Hitos del proyecto

5.4.2 Diagrama de Gantt

El diagrama de Gantt nos permite, gráficamente, revisar las tareas y actividades, su duración y fechas propuestas de cada una de ellas. Hay que considerar que las fechas presentadas son preliminares.

En la siguiente tabla se presentan las tareas resumidas y sus fechas.

Nombre de tarea	Duración	Meses				
		dic-20	ene-21	feb-21	mar-21	abr-21
INICIO	**4 días**					
PLANIFICACION	**15 días**					
Alcance del proyecto	**10 días**					
Gestión del Cronograma	**3 días**					
EJECUCION	**73 días**					
Estudio Teórico	**14 días**					
Metodología de trabajo	**2 días**					
Análisis de datos	**57 días**					
Preprocesamiento de datos	**30 días**					
Implementación de algoritmos	**15 días**					
Resultados	**10 días**					
SEGUIMIENTO	**80,13 días**					
CIERRE	**2 días**					
Acta de Cierre	2 días					
DURACIÓN TOTAL	**94 días**					

Tabla 4: Diagrama de Gantt

6 APLICACIÓN METODOLÓGICA

6.1 Resumen

Basándose en la metodología de trabajo de análisis de datos KDD dentro de los pasos a seguir tenemos que:

Paso 1: Limpieza y preprocesamiento de datos

- Análisis de datos
  - ✓ Análisis 1D
  - ✓ Análisis 2D

Paso 2: Reducción y proyección de datos

- Preprocesamiento de datos

Paso 3: Elección de los algoritmos de minería de datos

- Implementación de Algoritmos

Paso 4: Minería de datos

- Resultados de aplicación del modelo

Paso 5: Interpretación

- Experimentación del uso de Weka

Paso 6: Usar el conocimiento descubierto

- Análisis comparativo del modelo y Weka
- Conclusiones

6.2 Análisis de Datos

La muestra de datos a analizar consiste en información de 467 estudiantes clasificada en 80 atributos. Luego de eliminar 32 de estos atributos por ser considerados redundantes, o de formato alfanumérico no correspondiendo a categorías, o que no aportan algún tipo de información adicional al modelo, quedan disponibles 48 atributos. En la tabla 5 se

muestran los atributos que son descartados del análisis de datos y posterior generación del modelo. Los atributos “Tiempo de inicio” y “Tiempo de término”, de elaboración de la solución en cada nivel, son eliminados debido a que existe también el atributo “Tiempo total” para cada nivel, el cual corresponde a la sustracción de los dos atributos anteriores. Los atributos llamados “Solución generada” son descartados porque además de contener caracteres y no ser de tipo categóricos, se cuenta también con el atributo “Largo de solución generada”, el cual aporta información de tipo numérica y cuantitativa acerca de las soluciones.

Variables	Descripción	Tipos de datos
rut	Número correlativo	Int64
usuario	Anonimizado	N/A
correo	Anonimizado	N/A
nombre	Anonimizado	N/A
profesor	Anonimizado	N/A
i1	Tiempo de inicio nivel 1	Float64
f1	Tiempo de fin nivel 1	Float64
s1	Solución generada nivel 1	String
i2	Tiempo de inicio nivel 2	Float64
f2	Tiempo de fin nivel 2	Float64
s2	Solución generada nivel 2	String
i3	Tiempo de inicio nivel 3	Float64
f3	Tiempo de fin nivel 3	Float64
s3	Solución generada nivel 3	String
i4	Tiempo de inicio nivel 4	Float64
f4	Tiempo de fin nivel 4	Float64
s4	Solución generada nivel 4	String
i5	Tiempo de inicio nivel 5	Float64
f5	Tiempo de fin nivel 5	Float64
s5	Solución generada nivel 5	String
i6	Tiempo de inicio nivel 6	Float64
f6	Tiempo de fin nivel 6	Float64
s6	Solución generada nivel 6	String
sol1	Nota prueba solemne 1	Float64
sol2	Nota prueba solemne 2	Float64
sol3	Nota prueba solemne 3	Float64
sol4	Nota prueba solemne 4	Float64
tarea1	Nota tarea 1	Float64

tarea2	Nota tarea 2	Float64
controles	Nota de participación	Float64
np	Nota de presentación a examen	Float64
examen	Nota del examen	Float64

Tabla 5: Lista de atributos descartadas del análisis

Los 48 atributos considerados para el análisis, que de ahora en adelante llamaremos variables, pueden ser agrupados, para facilitar su entendimiento, según están relacionados al juego de diagnóstico previo al curso, la prueba de diagnóstico de aptitudes y el curso de programación. En la tabla 6 pueden verse las variables relacionadas al juego de diagnóstico.

Variables	Descripción	Tipo de datos
t1	Tiempo total nivel 1	float64
r1	Cantidad de reinicios del nivel 1	int64
p1	Cantidad de pruebas del nivel 1	int64
l1	Largo de la solución generada en el nivel 1	int64
t2	Tiempo total nivel 2	float64
r2	Cantidad de reinicios del nivel 2	int64
p2	Cantidad de pruebas del nivel 2	int64
l2	Largo de la solución generada en el nivel 2	int64
t3	Tiempo total nivel 3	float64
r3	Cantidad de reinicios del nivel 3	int64
p3	Cantidad de pruebas del nivel 3	int64
l3	Largo de la solución generada en el nivel 3	int64
t4	Tiempo total nivel 4	float64
r4	Cantidad de reinicios del nivel 4	int64
p4	Cantidad de pruebas del nivel 4	int64
l4	Largo de la solución generada en el nivel 4	int64
t5	Tiempo total nivel 5	float64
r5	Cantidad de reinicios del nivel 5	int64
p5	Cantidad de pruebas del nivel 5	int64
l5	Largo de la solución generada en el nivel 5	int64
t6	Tiempo total nivel 6	float64
r6	Cantidad de reinicios del nivel 6	int64
p6	Cantidad de pruebas del nivel 6	int64
l6	Largo de la solución generada en el nivel 6	int64
op1	¿Encontró solución óptima en el nivel 1?	int64
op2	¿Encontró solución óptima en el nivel 2?	int64

op3	¿Encontró solución óptima en el nivel 3?	int64
op4	¿Encontró solución óptima en el nivel 4?	int64
op5	¿Encontró solución óptima en el nivel 5?	int64
op6	¿Encontró solución óptima en el nivel 6?	int64
sv1	¿Usó más de las instrucciones permitidas en el nivel 1?	int64
sv2	¿Usó más de las instrucciones permitidas en el nivel 2?	int64
sv3	¿Usó más de las instrucciones permitidas en el nivel 3?	int64
sv4	¿Usó más de las instrucciones permitidas en el nivel 4?	int64
sv5	¿Usó más de las instrucciones permitidas en el nivel 5?	int64
sv6	¿Usó más de las instrucciones permitidas en el nivel 6?	int64
lt	Largo acumulado de las soluciones de los niveles	int64
tt	Tiempo total acumulado de los niveles	float64
pt	Total de pruebas acumuladas de los niveles	int64
game_score	Suma de lt + tt + pt	float64
programa	Carrera del estudiante	String
final	Nota final del estudiante	float64
estado	Estado final del estudiante (Aprobado / Reprobado)	int32

Tabla 6: Variables relacionadas al juego de diagnóstico

Respecto a este conjunto de variables es relevante la variable “game_score”, debido a que resume el desempeño del estudiante en esta prueba a partir de los resultados en los seis niveles de evaluación. Las variables booleanas “op1”, “op2”, “op3”, “op4”, “op5”, “op6”, “sv1”, “sv2”, “sv3”, “sv4”, “sv5” y “sv6” tienen asignados el valor “1” para el caso positivo y “0” para el caso negativo.

En la tabla 7 son mostradas las variables relacionadas con la prueba de diagnóstico. La variable “score” tiene un rango entre 1 y 13, y representa el resultado final de la evaluación a los estudiantes respecto a aptitudes consideradas necesarias para la programación.

Variables	Descripción	Tipo de datos
score	Puntaje total	int64
score_a	Puntaje de abstracción	int64
score_p	Puntaje de reconocimiento de patrones	int64
score_d	Puntaje de descomposición	int64
score_s	Puntaje de algoritmos	int64

Tabla 7: Variables relacionadas con la prueba de diagnóstico

En la tabla 8 se presentan las variables relacionadas con el curso de programación. Aquí cabe destacar la variable “estado”, la cual representa el hecho de si el estudiante aprobó o reprobó el curso de programación, por lo tanto, es una variable booleana para la que el valor “Aprobado” se reemplaza con “0” y el valor “Reprobado” con “1”. Esta variable estaría ubicada en la capa de salida del grafo generado, puesto que es la consecuencia final dentro de las inferencias que se quieren realizar en el modelo causal.

Variables	Descripción	Tipo de datos
programa	Carrera del estudiante	String
final	Nota final del estudiante	float64
estado	Estado final del estudiante (Aprobado / Reprobado)	String

Tabla 8: Variables relacionadas al curso de programación

La variable “programa” es de tipo categórica, ya que se refiere a la carrera que está estudiando el alumno en la universidad; por lo tanto, sus valores alfabéticos son reemplazados por un equivalente numérico. En la tabla 9 se muestra esta conversión.

Variable programa	Nuevo Valor
BACHILLERATO EN CIENCIAS	1
INGENIERIA INDUSTRIAL	2
INGENIERIA CIVIL INFORMATICA	3
INGENIERIA EN COMPUTACION E INFORMATICA	4
INGENIERIA CIVIL INDUSTRIAL	5

Tabla 9: Conversión de valores en variable “programa”

La variable “estado”, que también es de tipo categórica, y que tiene relación a la situación del curso, sus valores alfabéticos son reemplazados por un equivalente numérico. En la tabla 10 se muestra esta conversión.

Variable estado	Nuevo Valor
A	0
R	1

Tabla 10: Conversión de valores en variable "estado"

6.2.1 Análisis 1D

Para la realización del análisis unidimensional de los datos se utilizó el ambiente de trabajo llamado Anaconda Navigator, que permite crear ambientes de trabajo en Python utilizando los paquetes más utilizados para el desarrollo de proyectos de ciencia de datos, dentro del navegador anaconda se ejecuta Jupyter Notebook. Dentro de Jupyter se sube la base de datos que se encuentra en formato CSV y se guarda en un objeto para su posterior manejo. En la instrucción de lectura del archivo utiliza la librería *Pandas*.

En la figura 9 se muestra la instrucción de lectura.

```
In [2]:
#abriendo el conjunto de datos desde un archivo CSV y asignandolo los datos al DataFrame "df"
df = pd.read_csv('dataset_a.csv', sep=';', error_bad_lines=False)
```

Figura 9: Lectura de la base de datos

Luego se verifica que no existan valores nulos en los registros. En la figura 10 se puede verificar que no existen variables que posean valores nulos, lo que nos indica que no se requiere hacer reemplazo de valores faltantes. En caso de necesitarse, existen técnicas que permiten hacer el reemplazo de los valores nulos con la media existente.

```
In [4]:
#identificando las columnas con valores nulos "NaN"
null_columns=df.columns[df.isnull().any()]
df[null_columns].isnull().sum()
print(df[df.isnull().any(axis=1)][null_columns].head())

Empty DataFrame
Columns: []
Index: []
```

Figura 10: Verificación de valores nulos

Posteriormente, se ejecuta la instrucción para realizar un análisis descriptivo de los datos. Observando los valores resultantes para las primeras 15 variables en la figura 11, puede concluirse que los datos relacionados con los primeros 4 niveles del juego diagnóstico, poseen una alta dispersión y valores extremos alejados del promedio.

```
df.describe()
```

	tiempo1	resets1	pruebas1	largosol1	tiempo2	resets2	pruebas2	largosol2	tiempo3	resets3	pruebas3	largosol3	tiempo4	resets4	pruebas4
count	467.000000	467.000000	467.000000	467.000000	467.000000	467.000000	467.000000	467.000000	467.000000	467.000000	467.000000	467.000000	467.000000	467.000000	467.000000
mean	101.104817	2.179872	12.370450	8.856531	90.699059	1.897216	10.758030	11.417559	169.270350	2.199143	20.254818	21.027837	83.612827	1.554604	9.402570
std	110.090207	2.675148	20.590161	3.801227	92.952985	1.899326	11.804347	3.201923	117.380853	1.826415	19.320800	8.217358	82.956609	1.081802	11.493637
min	14.947463	0.000000	1.000000	7.000000	17.714780	1.000000	2.000000	10.000000	27.226233	1.000000	2.000000	16.000000	18.537399	1.000000	1.000000
25%	41.700748	1.000000	2.000000	7.000000	46.058281	1.000000	5.000000	10.000000	88.981408	1.000000	9.000000	16.000000	45.034990	1.000000	4.000000
50%	65.350279	1.000000	7.000000	7.000000	67.506870	1.000000	9.000000	10.000000	131.983587	1.000000	15.000000	17.000000	64.044106	1.000000	6.000000
75%	116.476445	2.000000	12.500000	9.000000	104.718704	2.000000	12.000000	12.000000	214.802128	3.000000	25.000000	23.000000	98.520291	2.000000	12.000000
max	962.077675	26.000000	214.000000	44.000000	1092.332500	24.000000	124.000000	29.000000	1080.815802	14.000000	208.000000	64.000000	1317.984673	11.000000	155.000000

Figura 11: Análisis 1D de las primeras 15 variables

En la figura 12 puede observarse para las siguientes 14 variables, que en los niveles 5 y 6 del juego diagnóstico los datos mantienen esta característica de alta dispersión. Las variables booleanas “optima1” y “optima2” muestran una tendencia de los estudiantes a encontrar la solución óptima en los niveles 1 y 2, sin embargo, en los niveles 3, 4 y 5 parece haber predominado el hecho de no haber encontrado la solución óptima.

largosol4	tiempo5	resets5	pruebas5	largosol5	tiempo6	resets6	pruebas6	largosol6	optima1	optima2	optima3	optima4	optima5
467.000000	467.000000	467.00000	467.000000	467.000000	467.000000	467.000000	467.000000	467.000000	467.000000	467.000000	467.000000	467.000000	467.000000
12.839400	63.107647	1.51606	6.805139	11.158458	73.065997	1.591006	8.811563	12.985011	0.556745	0.698073	0.398287	0.473233	0.411135
4.168901	43.064858	1.05902	6.387566	3.726219	77.560509	1.576982	16.704797	4.965308	0.497302	0.459586	0.490070	0.499818	0.492567
10.000000	15.399420	1.00000	1.000000	8.000000	16.305861	1.000000	1.000000	10.000000	0.000000	0.000000	0.000000	0.000000	0.000000
10.000000	38.392791	1.00000	3.000000	8.000000	41.732285	1.000000	4.000000	10.000000	0.000000	0.000000	0.000000	0.000000	0.000000
11.000000	52.555673	1.00000	5.000000	10.000000	56.591122	1.000000	7.000000	11.000000	1.000000	1.000000	0.000000	0.000000	0.000000
14.000000	73.212156	2.00000	8.000000	13.000000	83.588994	2.000000	10.000000	14.000000	1.000000	1.000000	1.000000	1.000000	1.000000
36.000000	527.017865	10.00000	61.000000	42.000000	1448.764917	19.000000	342.000000	49.000000	1.000000	1.000000	1.000000	1.000000	1.000000

Figura 12: Análisis 1D de las siguientes 14 variables

Respecto a las siguientes 14 variables, observando la figura 13 puede concluirse que en los niveles 1, 2, 4, 5 y 6 del juego existe la tendencia en los estudiantes de haber utilizado más instrucciones que las permitidas en la construcción de la solución, mientras que en el nivel 3 la tendencia es contraria. El puntaje final del juego representado en la variable “game_score”, al ser calculado a partir de los datos en los niveles, mantiene la alta dispersión y valores extremos, aunque parece haber disminuido respecto a los resultados

individuales de los niveles. Esta dispersión puede deberse a que exista heterogeneidad en el grupo de estudiantes.

optima6	sv1	sv2	sv3	sv4	sv5	sv6	largototal	tiempototal	pruebastotal	game_score	score_diag	score_abs	score_pat
467.000000	467.000000	467.000000	467.000000	467.000000	467.000000	467.000000	467.000000	467.000000	467.000000	467.000000	467.000000	467.000000	467.000000
0.479657	0.978587	0.942184	0.398287	0.760171	0.873662	0.796574	78.284797	580.860696	68.402570	18.531029	7.077088	4.404711	3.199143
0.500122	0.144913	0.233645	0.490070	0.427437	0.332587	0.402978	18.204084	286.479239	51.589755	6.872696	2.156547	1.546497	1.323670
0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	61.000000	177.997170	11.000000	9.682934	1.000000	0.000000	0.000000
0.000000	1.000000	1.000000	0.000000	1.000000	1.000000	1.000000	66.000000	384.790510	37.000000	13.995174	6.000000	3.000000	2.000000
0.000000	1.000000	1.000000	0.000000	1.000000	1.000000	1.000000	73.000000	509.651843	55.000000	17.100026	7.000000	4.000000	3.000000
1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	85.000000	700.930958	83.500000	20.946877	9.000000	5.500000	4.000000
1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	191.000000	2262.968067	623.000000	84.472331	12.000000	8.000000	5.000000

Figura 13: Análisis 1D de las siguientes 14 variables

Las variables relacionadas con la prueba diagnóstico, “score_diag”, “score_abs” y score_pat”, a pesar de mostrar valores extremos distantes del promedio, parecen tener una baja dispersión y distribución simétrica, dado que la media y mediana tienden a coincidir. Esto puede significar que la muestra de datos representa un grupo de estudiantes suficientemente completo en cuanto a variedad de aptitudes para la programación.

En la figura 14 que muestra las últimas 14 variables, los resultados relacionados con la prueba diagnóstico, en las variables “score_desc” y “score_alg”, muestran el mismo patrón de las primeras variables de esta prueba. Las notas de las solemnes tienden a ser altas en la mayoría de los alumnos, mientras que las notas de las tareas tienden a ser menores a las solemnes. Esto puede ser consecuencia de una mayor dificultad en la evaluación de las tareas, o puede existir una menor disposición de los estudiantes al trabajo fuera de aula.

score_desc	score_alg	programa	solemne1	solemne2	solemne3	solemne4	tarea1	tarea2	controles	npresent	examen	final	estado
467.000000	467.000000	467.000000	467.000000	467.000000	467.000000	467.000000	467.000000	467.000000	467.000000	467.000000	467.000000	467.000000	467.000000
2.197002	2.222698	2.252677	5.945824	5.620771	5.475161	5.659957	4.752248	4.605782	5.989293	5.297088	5.394861	5.331049	0.104925
1.033636	0.875071	1.127289	1.161433	1.306757	1.444950	1.148895	2.272444	2.338002	1.844250	1.280340	1.454686	1.287560	0.306785
0.000000	0.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	0.000000
1.000000	2.000000	1.000000	5.600000	5.200000	5.000000	5.300000	3.000000	2.500000	5.900000	4.730000	5.100000	4.900000	0.000000
2.000000	2.000000	2.000000	6.300000	6.000000	6.000000	6.000000	5.800000	5.400000	7.000000	5.610000	5.700000	5.600000	0.000000
3.000000	3.000000	3.000000	6.800000	6.400000	6.400000	6.400000	6.800000	6.900000	7.000000	6.215000	6.300000	6.200000	0.000000
4.000000	4.000000	5.000000	7.000000	7.000000	7.000000	7.000000	7.000000	7.000000	7.000000	7.000000	7.000000	7.000000	1.000000

Figura 14: Análisis 1D de las últimas 14 variables

Adicionalmente, se calcula la frecuencia de los valores en la variable categórica “programa” y la variable booleana “estado”, ya que el análisis anterior no parece ser muy adecuado para estas variables. La figura 15 muestra los resultados del cálculo representados en porcentajes, éstos indican que una cantidad cercana a la mitad de los estudiantes cursa la carrera de Ingeniería Civil Industrial [5] (40.26%). Además, el índice de reprobación en el curso de programación es bastante bajo respecto a los aprobados (10.49%).

Figura 15: Cálculo de porcentaje para las variables “programa” y “estado"

6.2.2 Análisis 2D

Para el análisis bidimensional de las variables cuantitativas, se seleccionan diez de ellas consideradas las más representativas y/o correlacionadas, luego se genera una matriz de correlación cuyas celdas contienen gráficos de dispersión. Las variables elegidas son “lt”, “tt”, “pt”, “game_score”, “score”, “score_a”, “score_p”, “score_d”, “score_s” y “final”.

En la figura 16 se muestra la matriz de correlación, donde puede observarse en primer lugar, una correlación positiva entre las variables “lt” y “game_score”, entre “tt” y “game_score”, y entre “pt” y “game_score”, siendo las dos primeras bajas y la tercera mayor respecto a las anteriores. Además, no parece haber correlación entre “lt”, “tt” y “pt”. Esto quiere decir que, respecto al juego diagnóstico el largo total de las soluciones, el tiempo total invertido y la cantidad total de pruebas individualmente tienden a aumentar cuando el puntaje total aumenta, pero entre ellas esta tendencia no existe.

También se encuentra una correlación positiva entre las variables “score” con respecto a “score_a”, “score_p”, “score_d”, “score_s”

Los histogramas muestran su dispersión en función de la variable “programa”

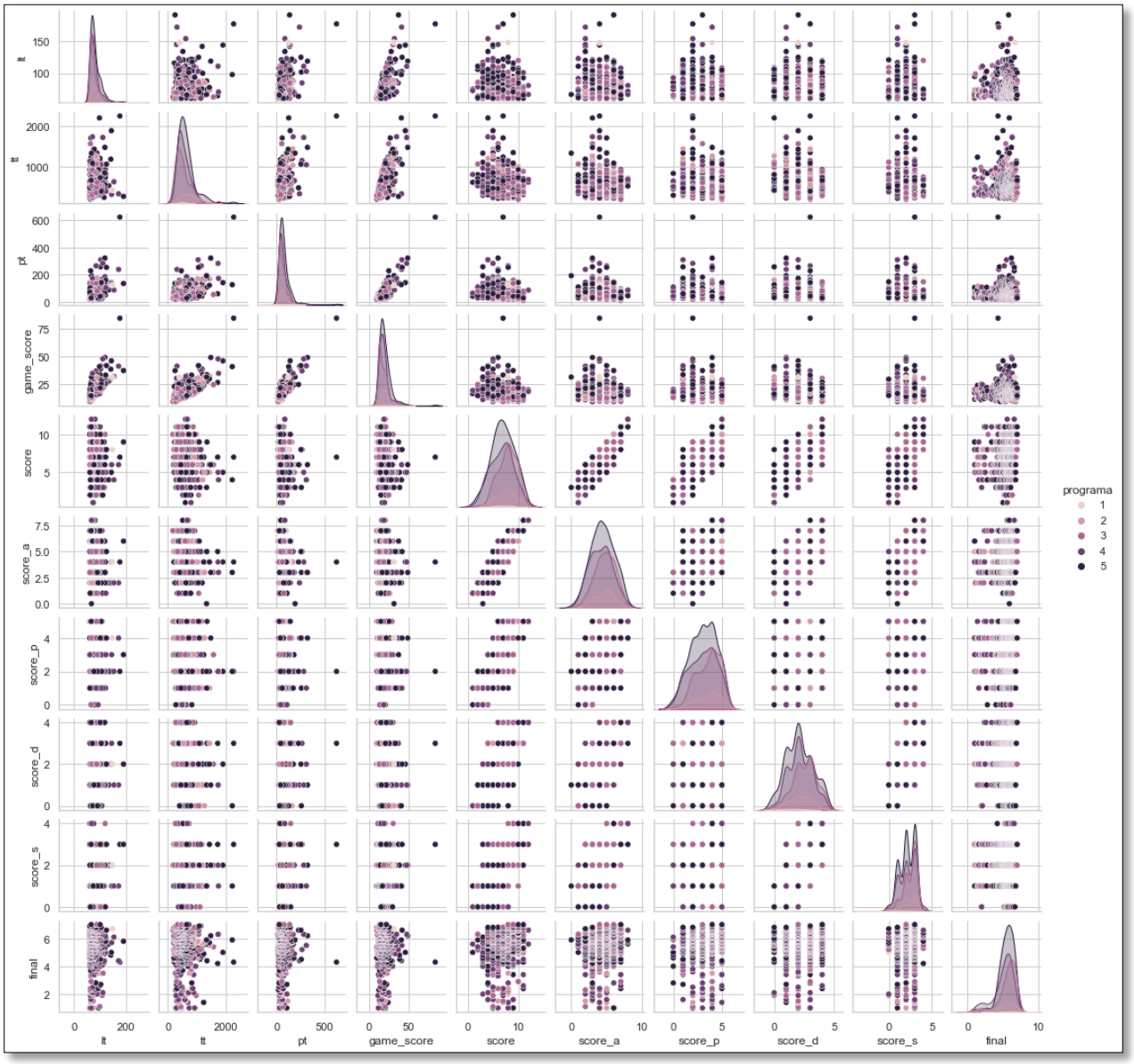

Figura 16: Matriz de correlación de variables

Para las variables categóricas “op1”, “op2”, “op3”, “op4”, “op5”, “op6”, “sv1”, “sv2”, “sv3”, “sv4”, “sv5”, “sv6”, “programa” y “estado”, se realizan tablas de contingencia.

En la figura 17 se muestra el cruce entre las variables “programa” y “estado”, donde puede observarse que, la carrera de Ingeniería Civil Industrial [5] tiene el más bajo porcentaje de reprobación sin contar Bachillerato en Ciencias [1] dado que la muestra no es representativa (solo existen dos alumnos en esa carrera en la base de datos).

```
In [15]:
#Se obtiene la tabla de contingencia de "programa" vs "estado" obteniendo la normalización
#de los datos respecto a todos los programas
pd.crosstab(df.programa, df.estado, normalize='all', margins=True)\
    .round(4)*100
```

Out[15]:

estado	0	1	All
programa			
1	0.43	0.00	0.43
2	5.14	1.50	6.64
3	21.20	2.14	23.34
4	24.41	4.93	29.34
5	38.33	1.93	40.26
All	89.51	10.49	100.00

Figura 17: Tabla de contingencia programa / estado

En las figuras 18 al 23 se muestran las tablas de contingencia para las variables “op1” y “sv1” al 6. En la primera se puede observar que los estudiantes que encontraron la solución óptima en el nivel 1 usaron más de las instrucciones permitidas. En el cruce de las variables “op2” y “sv2” se repite la tendencia del cruce anterior, pero en el cruce entre las variables “op3” y “sv3” la tendencia se rompe dado que los alumnos encontraron la solución óptima con menos instrucciones permitidas, retomando la tendencia para los niveles 4, 5 y 6.

```
Se analiza la tabla de contigencia de dos variables, para los 6 niveles del juego de dignóstico

op(x) = "¿Encontró solución optima en el Nivel 1?", dónde 0 = NO y 1 = SÍ
sv(x) = "¿Usó más de las instrucciones permitidas?", , dónde 0 = NO y 1 = SÍ
```

```
In [16]:
#Se obtiene la tabla de contingencia de "op1" vs "sv1" obteniendo la normalización
#de los datos respecto a todos los que encontraron o no soliciones optimas
pd.crosstab(df.op1, df.sv1, normalize='all', margins=True)\
    .round(4)*100
```

Out[16]:

sv1	0	1	All
op1			
0	2.14	42.18	44.33
1	0.00	55.67	55.67
All	2.14	97.86	100.00

Figura 18: Tabla de contingencia op1 / sv1

```
In [17]:
#Se obtiene la tabla de contingencia de "op2" vs "sv2" obteniendo la normalización
#de los datos respecto a todos los que encontraron o no soliciones optimas
pd.crosstab(df.op2, df.sv2, normalize='all', margins=True)\
    .round(4)*100
```

Out[17]:

sv2	0	1	All
op2			
0	5.78	24.41	30.19
1	0.00	69.81	69.81
All	5.78	94.22	100.00

Figura 19: Tabla de contingencia op2 / sv2

```
In [18]:
#Se obtiene la tabla de contingencia de "op3" vs "sv3" obteniendo la normalización
#de los datos respecto a todos los que encontraron o no soliciones optimas
pd.crosstab(df.op3, df.sv3, normalize='all', margins=True)\
    .round(4)*100
```

Out[18]:

sv3	0	1	All
op3			
0	60.17	0.00	60.17
1	0.00	39.83	39.83
All	60.17	39.83	100.00

Figura 20: Tabla de contingencia op3 / sv3

```
In [19]:
#Se obtiene la tabla de contingencia de "op4" vs "sv4" obteniendo la normalización
#de los datos respecto a todos los que encontraron o no soliciones optimas
pd.crosstab(df.op4, df.sv4, normalize='all', margins=True)\
    .round(4)*100
```

Out[19]:

sv4	0	1	All
op4			
0	23.98	28.69	52.68
1	0.00	47.32	47.32
All	23.98	76.02	100.00

Figura 21: Tabla de contingencia op4 / sv4

```
In [20]:
#Se obtiene la tabla de contingencia de "op5" vs "sv5" obteniendo la normalización
#de los datos respecto a todos los que encontraron o no soliciones optimas
pd.crosstab(df.op5, df.sv5, normalize='all', margins=True)\
    .round(4)*100
```

Out[20]:

sv5	0	1	All
op5			
0	12.63	46.25	58.89
1	0.00	41.11	41.11
All	12.63	87.37	100.00

Figura 22: Tabla de contingencia op5 / sv5

```
In [21]:
#Se obtiene la tabla de contingencia de "op6" vs "sv6" obteniendo la normalización
#de los datos respecto a todos los que encontraron o no soliciones optimas
pd.crosstab(df.op6, df.sv6, normalize='all', margins=True)\
    .round(4)*100
```

Out[21]:

sv6	0	1	All
op6			
0	20.34	31.69	52.03
1	0.00	47.97	47.97
All	20.34	79.66	100.00

Figura 23: Tabla de contingencia op6 / sv6

6.3 Preprocesamiento de datos

Para el preprocesamiento de los datos se realizó una selección de las variables relevantes en función de los análisis 1D y 2D previamente vistos. Dichas variables fueron las que mostraron una mayor correlación entre sí, indicando un mayor aporte de información con respecto a las demás variables. Las variables seleccionadas fueron las siguientes:

Variables	Descripción	Tipo de datos
op1	¿Encontró solución óptima en el nivel 1?	int64
op2	¿Encontró solución óptima en el nivel 2?	int64
op3	¿Encontró solución óptima en el nivel 3?	int64
op4	¿Encontró solución óptima en el nivel 4?	int64
op5	¿Encontró solución óptima en el nivel 5?	int64
op6	¿Encontró solución óptima en el nivel 6?	int64
sv1	¿Usó más de las instrucciones permitidas en el nivel 1?	int64
sv2	¿Usó más de las instrucciones permitidas en el nivel 2?	int64
sv3	¿Usó más de las instrucciones permitidas en el nivel 3?	int64
sv4	¿Usó más de las instrucciones permitidas en el nivel 4?	int64
sv5	¿Usó más de las instrucciones permitidas en el nivel 5?	int64
sv6	¿Usó más de las instrucciones permitidas en el nivel 6?	int64
lt	Largo acumulado de las soluciones de los niveles	int64
tt	Tiempo total acumulado de los niveles	float64
pt	Total de pruebas acumuladas de los niveles	int64
game_score	Suma de lt + tt + pt	float64
score	Puntaje total	int64
score_a	Puntaje de abstracción	int64
score_p	Puntaje de reconocimiento de patrones	int64
score_d	Puntaje de descomposición	int64
score_s	Puntaje de algoritmos	int64

programa	Carrera del estudiante	String
final	Nota final del estudiante	float64
estado	Estado final del estudiante (Aprobado / Reprobado)	int32

Tabla 11: Listado de variables seleccionadas para realizar el modelo

Para el preprocesamiento de datos se realizó la discretización de algunas variables dado que estos poseen datos continuos que no permitirían ejecutar de buena manera el modelo que se implementa en el siguiente paso. Las variables que fueron discretizadas fueron las siguientes:

Variables	Descripción	Tipo de datos
lt	Largo acumulado de las soluciones de los niveles	int64
tt	Tiempo total acumulado de los niveles	float64
pt	Total de pruebas acumuladas de los niveles	int64
game_score	Suma de lt + tt + pt	float64
final	Nota final del estudiante	float64

Tabla 12: Lista de variables discretizadas

Para la discretización de las variables se usó la función de discretización *KBinsDiscretizer* de la librería *sklearn* (Pedregosa, y otros, 2011) que permite transformar datos continuos en intervalos dependiendo de la cantidad de "bins" requeridos. Para nuestro caso, las variables "lt", "tt", "pt" y "game_score" se usaron 5 "bins" en la discretización y en la variable "final" se usó una discretización de 7 "bins". Al momento de la discretización de las variables se usó una estrategia algorítmica llamada "k-means" que corresponde a una clasificación no supervisada que agrupa objetos en k grupos basándose en sus características.

6.4 Implementación de algoritmos

En esta fase se construyó un algoritmo que permite realizar el aprendizaje de la estructura y de parámetros basándose en la librería *bnlearn* (Taskesen, bnlearn, 2019). El algoritmo consiste en realizar un muestreo estratificado usando como clase la variable "estado". Para la estratificación de los datos se usó la librería *sklearn* (Pedregosa, y otros, 2011), en particular la función *StratifiedKFold* que proporciona una división de los datos en conjuntos de entrenamiento y pruebas, donde se le indica el número de experimentos

que se desea usar. Para el caso del algoritmo implementado se usaron 5 experimentos que permite realizar el aprendizaje bayesiano que es usado para inferir las variables usando una probabilidad conjunta. En cada experimento se aprende y se infiere, tanto para la partición de entrenamiento como para la porción de pruebas, para obtener una puntuación (accuracy), una puntuación balanceada (balanced accuracy) y una ratio del modelo implementado.

A continuación, se presenta el modelo algorítmico implementado:

1. estratificar usando 5 split barajando las muestras antes de dividirlas.
2. Inicializar contador de Split en 1
3. Para índice train y test dentro del Split de la clase "estado" hacer,
   a. Obtener porción de datos de training
   b. Aprender su estructura y sus parámetros
   c. Obtener probabilidad conjunta
   d. Obtener porción de datos de testing
   e. Aprender su estructura y sus parámetros
   f. Obtener probabilidad conjunta
   g. Obtener puntuación de precisión, puntuación equilibrada y la proporción para la porción de training
   h. Obtener puntuación de precisión, puntuación equilibrada y la proporción para la porción de testing
   i. Aumentar contador de Split

6.5 Requerimientos Técnicos

Como primer paso para manejar la base de datos se recomienda utilizar como plataforma Anaconda Navigator, que nos permite trabajar con el lenguaje de ciencia de datos Python. Una vez instalada esta plataforma se puede crear un ambiente con las herramientas y librerías necesarias para la correcta ejecución del modelo construido. Los siguientes son las librerías y herramientas necesarias:

- Entorno de desarrollo JupyterLab: Como entorno de desarrollo interactivo basado en web para notebook, código y datos de código abierto que permite crear y compartir documentos que contienen código en vivo, ecuaciones, visualizaciones y texto narrativo. (Jupyter, 2021)
- Librerías utilizadas:
    a) **Pandas**: Biblioteca de código abierto con licencia BSD que proporciona estructuras de datos y herramientas de análisis de datos de alto rendimiento y fáciles de usar para el lenguaje de programación Python. (team, 2020)
    b) **Numpy**: Es el paquete fundamental para la computación científica en Python. Es una biblioteca de Python que proporciona un objeto de matriz multidimensional, varios objetos derivados (como matrices y matrices enmascaradas) y una variedad de rutinas para operaciones rápidas en matrices, que incluyen manipulación matemática, lógica, de formas, clasificación, selección, E / S., transformadas discretas de Fourier, álgebra lineal básica, operaciones estadísticas básicas, simulación aleatoria y mucho más. NumPy es un proyecto de código abierto que tiene como objetivo permitir la computación numérica con Python. Fue creado en 2005, basándose en el trabajo inicial de las bibliotecas Numeric y Numarray. NumPy siempre será un software 100% de código abierto, de uso gratuito para todos y publicado bajo los términos liberales de la licencia BSD modificada. (Harris, y otros, 2020)
    c) **Matplotlib**: Es una biblioteca para hacer gráficos 2D de matrices en Python. Aunque tiene su origen en la emulación de los comandos gráficos de MATLAB, es independiente de MATLAB y se puede utilizar de forma Pythonic y orientada a objetos. Aunque Matplotlib está escrito principalmente en Python puro, hace un uso intensivo de NumPy y otros códigos de extensión para proporcionar un buen rendimiento incluso para matrices grandes. (Hunter, 2007)
    d) **Seaborn**: Es una biblioteca de visualización de datos de Python basada en matplotlib. Proporciona una interfaz de alto nivel para dibujar gráficos estadísticos atractivos e informativos. (Waskom, 2021)

e) **Sklearn**: Es una biblioteca de aprendizaje automático cada vez más popular. Escrito en Python, está diseñado para ser simple y eficiente, accesible para no expertos y reutilizable en varios contextos (Buitinck, y otros, 2013)

f) **Bnlearn**: Paquete Python para aprender la estructura gráfica de redes bayesianas, aprendizaje de parámetros, inferencia y métodos de muestreo. (Taskesen, bnlearn, 2019)

g) **JSON**: Es un formato ligero de intercambio de datos. Leerlo y escribirlo es simple para humanos, mientras que para las máquinas es simple interpretarlo y generarlo. Está basado en un subconjunto del Lenguaje de Programación JavaScript, Standard ECMA-262 3rd Edition - diciembre 1999. JSON es un formato de texto que es completamente independiente del lenguaje, pero utiliza convenciones que son ampliamente conocidos por los programadores de la familia de lenguajes C, incluyendo C, C++, C#, Java, JavaScript, Perl, Python, y muchos otros. Estas propiedades hacen que JSON sea un lenguaje ideal para el intercambio de datos. (Pezoa, Reutter, Suarez, & Ugarte, 2016)

- Requerimientos de maquina: Se recomienda un requerimiento mínimo de un computador Intel i7 con una capacidad de 16GB de RAM y un disco duro de estado sólido. De ser posible contar con una tarjeta gráfica aparte, para ayudar a acelerar el procesamiento.

6.6 Resultados de aplicación del modelo

Para la obtención de los resultados se ejecutó el modelo descrito, con los requerimientos técnicos mencionados en el punto anterior, obteniendo un tiempo de respuesta de aproximadamente 8 horas de procesamiento.

A continuación, se muestra la tabla 13 con el resumen de los datos obtenidos:

Fold	Muestra	Accuracy	Balanced Accuracy	Ratio
1	Train	0,8659517426273450	0,5832919254658380	0,1045576407506700
	Test	0,8829787234042550	0,6172161172161170	0,1063829787234040
2	Train	0,8766756032171580	0,6212121212121210	0,1045576407506700

	Test	0,8936170212765950	0,8936170212765950	0,1063829787234040
3	Train	0,8930481283422460	0,6498644986449860	0,1042780748663100
	Test	0,9462365591397850	0,8514412416851440	0,1075268817204300
4	Train	0,0000000000000000	0,0000000000000000	0,1042780748663100
	Test	0,8924731182795690	0,7147058823529410	0,1075268817204300
5	Train	0,8689839572192510	0,6219980787704130	0,1069518716577540
	Test	0,8709677419354830	0,6927244582043340	0,0967741935483871

Tabla 13: Valores de precisión de los experimentos del modelo implementado

De la presente tabla se puede observar que el experimento 4 no obtuvo puntuación en la porción de datos correspondientes al entrenamiento, por consiguiente, este experimento no puede ser considerado para el promedio general. Los resultados completos de cada experimento pueden ser revisados en el repositorio de GitHub dispuesto para obtener el código fuente y los resultados previa solicitud a los autores.

https://github.com/rtarbes/tesisBN

En el repositorio se puede acceder a los resultados obtenidos del aprendizaje de parámetros donde se puede ver la probabilidad de distribución condicional (CPD). Además, en el apartado de anexos (punto 10.1) se pueden encontrar la trama de cada porción de datos y experimentos aplicados. Adicionalmente, en el repositorio se puede conocer la probabilidad conjunta de cada porción de datos y experimentos aplicados.

A continuación, se presenta la siguiente tabla que resume los experimentos válidos promediando los resultados obtenidos en la tabla 13

Muestra	$\overline{X}$ Accuracy	$\overline{X}$ Balanced Accuracy	$\overline{X}$ Ratio
Train	88%	62%	11%
Test	90%	76%	10%

Tabla 14: Promedio de precisión de los experimentos

6.7 Experimentación del uso de Weka

Se abre el archivo CSV de la muestra de datos en la herramienta Weka, luego se seleccionan las 24 variables más relevantes para el entrenamiento, después se aplican filtros para señalar las variables nominales y discretizar las variables continuas. La discretización se realiza con el método Equal Frequency usando 5 bins. Luego, se realiza

el entrenamiento dejando un 30% de la muestra para test, usando el algoritmo Hill Climber y el estimador simple. A continuación, se muestra un resumen de los resultados obtenidos que puede ser complementado con el anexo 10.2:

```
=== Summary ===
Correctly Classified Instances       126     90%
Incorrectly Classified Instances     14      10%
Kappa statistic                      0.3099
Mean absolute error                  0.0947
Root mean squared error              0.2306
Relative absolute error              53.9149%
Root relative squared error          87.8761%
Total Number of Instances            140


=== Detailed Accuracy By Class ===
TP Rate FP Rate Precision      Recall  F-Measure      MCC     ROC Area       PRC Area       Class
0,938   0,600   0,953          0,938   0,946          0,311   0,943          0,995          0
0,400   0,062   0,333          0,400   0,364          0,311   0,943          0,399          1

Weighted Avg.  0,900   0,562   0,909   0,900   0,904   0,311   0,943   0,952

=== Confusion Matrix ===
a       b       <--     classified as
122     8       |       a = 0
6       4       |       b = 1
```

7 CONCLUSIONES

El objetivo fundamental de la tesis era de abordar el problema de las posibles causas de reprobación del curso de programación de primer año del plan de estudio de los alumnos de la Universidad Andrés Bello y aportar una solución proactiva en la identificación de variables que permitan determinar las posibles causas de una reprobación del curso.

De acuerdo con el análisis de los objetivos específicos planteados, se exponen las siguientes conclusiones:

En relación con el primer objetivo, Preparar el 100% de la base de datos para su utilización, se puede concluir que esta se logró mediante las técnicas de análisis de datos utilizadas, como lo fueron, los análisis descriptivos y análisis 1D.

En lo que respecta al segundo objetivo, Identificar, como mínimo, un numero de 5 variables relevantes que permitan realizar el estudio, se puede concluir que estas fueron detectadas a través, de un análisis 2D, encontrando 10 variables que fueron las que poseen un mayor factor de correlación entre sí.

En cuanto al tercer objetivo, Evaluar y seleccionar las herramientas de software que proporcionen las funciones requeridas para el modelamiento, se puede concluir que se realizó un estudio de 2 librerías Python para el modelamiento de Redes Bayesianas (BNLearn y Pomegranate) del cual se determinó que BNLearn fue la más optima debido a la facilidad de uso, mayor documentación en internet, uso de variables continuas y discretas con respecto a su alternativa. Adicionalmente, se utilizó como herramienta software empaquetado, la llamada Weka, que permitió construir el modelo bayesiano y realizar una comparación de los resultados de ambos modelos.

En cuanto al cuarto objetivo, Modelar el problema de predicción usando una Red Bayesiana, que permita aplicar la utilización de las variables identificadas, se puede concluir que, efectivamente, se logró construir el modelo con las 10 variables mejor correlacionadas.

En cuanto al quinto objetivo, Generar métricas para evaluar la efectividad y desempeño del modelo generado, se puede concluir que se logró obtener métricas de precisión que

permitieron identificar que los datos con lo que se realizó el estudio y los valores de precisión encontrados permitieron deducir que el modelo necesita de ajustes, dado que la efectividad de este no es el esperado. Este tema se puede revisar en el apartado sobre los resultados del modelo bayesiano a continuación de este resumen.

Sobre el sexto objetivo planteado, Evaluar los resultados obtenidos para identificar las causas que desencadenan la reprobación, se puede concluir que los resultados obtenidos no permiten determinar con precisión que el modelo construido sea efectivo, por lo tanto, se deberán sugerir algunas técnicas para contrarrestar los problemas con clases desbalanceadas y mejorar la puntuación de este.

Para finalizar, la principal aportación de este trabajo consiste en la construcción de un modelo predictivo basado en Redes Bayesianas que sirva de herramienta proactiva al aprendizaje predictivo para la toma de decisiones con respecto al curso analizado.

7.1 Resultados del Modelo Bayesiano

Los resultados del modelo bayesiano construido han permitido determinar que la base de datos utilizada para el análisis está desequilibrada producto de que, del 100% de los datos utilizados, un 89,5% de estos corresponden a alumnos que aprobaron el curso de programación y solo un 10,5% de ellos reprobó. Por lo anterior, al aplicar el modelo bayesiano la precisión del modelo nos estará entregando, mayoritariamente, una respuesta de aprobación, por consiguiente, y debido a lo recién expuesto, se aplicó un método de precisión equilibrada para evaluar la exactitud del modelo utilizado. La precisión equilibrada funciona de forma diferente, ya que entrega una precisión media por clase donde el valor resultante más cercano a 1 indica que tan preciso es el modelo. La siguiente es la fórmula de la precisión equilibrada:

$Balanced\ Accuracy = \frac{\frac{TP}{(TP+FN)}+\frac{TN}{TN+FP}}{2}$, dónde $\frac{TP}{(TP-FN)}$ corresponde a la sensibilidad o tasa positiva verdadera y $\frac{TN}{(TN-FP)}$ corresponde a la especificidad o tasa negativa verdadera simplificando la formula como $Balanced\ Accuracy = \frac{\text{Sensitivity+Specificity}}{2}$.

Para el estudio realizado, se estratificaron los datos en 5 experimentos (fold) con una división en proporción a un 80/20 por ciento de los datos, donde el 80% de ellos se usaron para entrenamiento (train) y el 20% de ellos para pruebas (test). En términos reales, el 80% corresponde a 373 o 374 registros de entrenamiento por experimento y a 93 o 94 registros para pruebas. Los valores de exactitud nos indican que el modelo no es lo suficientemente preciso ya que los datos entregados con la porción de información de entrenamiento y pruebas indican un 62% y 76% en promedio respectivamente.

7.2 Resultados del Modelo Weka

La herramienta Weka, a pesar de las restricciones que puede poseer en comparación a otros métodos de análisis de datos, resulta ser una forma rápida y sencilla de pre-procesar datos, configurar un experimento, ejecutar el experimento y obtener resultados, evitando las dificultades que pueden surgir de la programación.

En cuanto al experimento de redes bayesianas realizado con esta herramienta, el grafo resultante muestra que efectivamente fueron encontradas relaciones entre las variables del modelo, estando siete de estas directamente relacionadas con la clase "estado": "score_diag", "score_abs", "score_pat", "score_desc", "score_alg", "programa" y final". De este hecho puede deducirse que las puntuaciones totales y parciales de la prueba diagnóstico y la carrera del estudiante, son características que pueden estar altamente relacionadas con el hecho de que el estudiante apruebe o repruebe el curso de programación.

Weka nos entrega varias medidas de rendimiento del modelo, entre ellas está la TP Rate, FP Rate, Precisión, Recall, F-Measure, MCC ROC. Con estas métricas podemos obtener una idea más global del modelo aplicado en esta herramienta.

El desglose es el siguiente, para terminar en un resumen que concluya la información:

- TP Rate: Esta medida, que corresponde a la tasa de verdaderos positivos o instancias correctamente clasificadas como una clase determinada, nos entrega un valor ponderado de 0,900 que es bastante cercano a 1 y que es considerado muy bueno.

- FP Rate: Esta medida, que corresponde a la tasa de falsos positivos o instancias clasificadas falsamente con una clase determinada, y nos entrega un promedio ponderado de 0,562, indicando una probabilidad media de error en la detección de que los valores reprobados (clase 1) se consideren aprobados (clase 0).
- Precision: Corresponde a la proporción de instancias que son verdaderamente de una clase, dividida por el total de instancias clasificadas como esa clase, donde el valor promedio ponderado corresponde a 0,909 que es bastante alta considerando que de 9 de 10 casos serian detectadas como aprobaciones (clase 0).
- Recall: Corresponde a la ratio entre los verdaderos positivos y la suma de los verdaderos positivos y los falsos negativos, es decir, la ratio entre los verdaderos positivos y los positivos reales. El valor promedio ponderado entregado corresponde a 0,900 lo que indica que casi todas las aprobaciones (clase 0) serían detectadas como verdaderos positivos.
- F-Measure: Es una medida que combina la precisión y la recuperación (recall) para devolver una medida de calidad más general del modelo. Para el caso la medida promedio ponderado entregado, corresponde a 0,904 indicando nuevamente una puntuación bastante alta.
- MCC ROC: Es uno de los valores más importantes emitidos por weka y dan una idea de cómo se están desempeñando los clasificadores, para el caso analizado por la herramienta, el valor promedio ponderado fue de 0,311 indicando que el modelo no tendrá un buen desempeño

En resumen, el modelo ejecutado con la herramienta Weka nos está indicando en casi todas las métricas que está perfecto, pero viendo el ultimo indicador nos proporciona una información valiosa de que el modelo tiene una inconsistencia que debe ser revisada u que debe estar originada en los datos desequilibrados con los que se cuenta ya que la proporción de los registros de aprobación (clase 0) versus los reprobados (clase 1) es muy marcada.

7.3 Análisis comparativo Modelo Bayesiano vs Weka

En el estudio realizado se construyó un modelo Bayesiano utilizando programación Python con las librerías estudiadas y seleccionadas para tal y también se desarrollo el mismo modelo con una herramienta existente llamada Weka, dónde, en ambos casos, se determinó que existe un sesgo con la base de datos utilizada ya que el desequilibrio entre las clases es muy marcado. Hablando en términos numéricos la proporción es de 89,5% de aprobaciones (clase 0) versus 10,5 de reprobaciones (clase 1).

Dada la proporción de información tan desigual, se recomienda realizar un estudio de técnicas que permitan analizar de una mejor manera las bases de datos que contienen datos desbalanceados.

8 REFERENCIAS BIBLIOGRÁFICAS

Aguirre, N. (2012). *Factores que predicen el rendimiento académico en la Escuela de Ingeniería de la Universidad de Chile.* Tesis de Magíster. Obtenido de http://repositorio.uchile.cl/handle/2250/112299

Azevedo, A., & Santos, M. (2008). KDD, SEMMA and CRISP-DM: a parallel overview. (I. P. Porto, Ed.) *ISCAP - Informática - Comunicações em eventos científicos*. Obtenido de http://hdl.handle.net/10400.22/136

Barahona U, P. (2014). Factores determinantes del rendimiento académico de los estudiantes de la Universidad de Atacama. *Estudios Pedagógicos (Valdivia), 40*(1), 25-39. Obtenido de https://dx.doi.org/10.4067/S0718-07052014000100002

Bayes, T. (31 de 12 de 1763). An Essay towards solving a Problem in the Doctrine of Chances. *53*.

Buitinck, L., Louppe, G., Blondel, M., Pedregosa, F., Mueller, A., Grisel, O., . . . Varoquaux, G. (2013). *{API} design for machine learning software: experiences from the scikit-learn.* Obtenido de https://arxiv.org/abs/1309.0238

Costa, E., Fonseca, B., Santana, M., Araújo, F., & Rego, J. (08 de 2017). Evaluating the effectiveness of educational data mining techniques for early prediction of students' academic failure in introductory programming courses. *73*, 247-256.

Fayyad, U., & Piatetsky-Shapiro, G. &. (1996). The KDD Process for Extracting Useful Knowledge from Volumes of Data. *Communications of the AC, 39*, 27-34.

Garbanzo, G. (2007). Factores asociados al rendimiento académico en estudiantes universitarios, una reflexión desde la calidad de la educación superior pública. *Universidad de Costa Rica. Educación, 31*(1), 43-63.

Harris, C., Millman, K., Van der Walt, S., Gommers, R., Virtanen, P., Cournapeau, D., . . . Oliphant, T. (09 de 2020). Array programming with NumPy. (S. S. Media, Ed.) *Nature, 585*(7825), 357-362. doi:10.1038/s41586-020-2649-2

He, Y., Chen, R., Li, X., Hao, C., Liu, S., Zhang, G., & Jiang, B. (2020). Online At-Risk Student Identification using RNN-GRU Joint Neural Networks. *11*, 474.

Hunter, J. (2007). Matplotlib: A 2D graphics environment. (I. C. SOC, Ed.) *Computing in Science & Engineering, 9*(3), 90-95. doi:10.1109/MCSE.2007.55

Institute, S. (1998). Data Mining and the Case for Sampling.

Jupyter, C. (02 de 04 de 2021). *Jupyter*. Obtenido de https://jupyter.org/

KDnuggets. (10 de 2014). *KDnuggets*. Obtenido de https://www.kdnuggets.com/2014/10/crisp-dm-top-methodology-analytics-data-mining-data-science-projects.html

López Balanzátegui, M. E., Flores Herrera, J., Flores Nicolalde, B., & Flores Nicolalde, F. (03 de 2016). Predicción del rendimiento académico de los estudiantes de física a través de las redes bayesianas en la unidad de cantidad de movimiento lineal. *Lat. Am. J. Phys. Educ., 10*(1), 1408-1/14.

Marco Galindo, M. J., Minguillón, J., & Sancho-Vinuesa, T. (2020). Análisis de la progresión de los estudiantes en una asignatura introductoria a la programación mediante redes bayesianas. *Actas de las Jenui, 5*, 69-76.

Minerva. (s.f.). Obtenido de https://mnrva.io/kdd-platform.html

Morales, M., & Salmerón, A. (08-11 de 04 de 2003). Análisis del alumnado de la Universidad de Almería mediante Redes Bayesianas. (D. d. Aplicada, Ed.) *27 Congreso Nacional de Estadística e Investigación Operativa*, 1-24.

Neapolitan, R. E. (2003). *Learning Bayesian Networks* (1 ed.). Prentice Hall.

Oviedo, B., Puris, A., Villacís, A., Delgado, D., & Moreno, A. (07 de 2015). Análisis de datos educativos utilizando Redes Bayesianas. Obtenido de https://www.researchgate.net/publication/282349044

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., . . . Duchesnay, E. (2011). Scikit-learn: Machine Learning in {P}ython. *Journal of Machine Learning Research, 12*, págs. 2825-2830.

Peralta, F. (2014). Proceso de Conceptualización del Entendimiento del Negocio para Proyectos de Explotación de Información.

Pezoa, F., Reutter, J., Suarez, F., & Ugarte, M. (2016). *Foundations of JSON schema.* Proceedings of the 25th International Conference on World Wide Web.

PMI. (s.f.). *PMBOK® Guide and Standards*. Obtenido de https://www.pmi.org/pmbok-guide-standards/standard-for-project-management-exposure-draft/changes-to-the-pmbok-guide-seventh-edition

Rodríguez Suárez, Y., & Díaz Amador, A. (2009). Herramientas de Minería de Datos. *Revista Cubana de Ciencias Informáticas, 3*(3-4), 73-80. Obtenido de https://www.redalyc.org/articulo.oa?id=3783/378343637009

Román, J. V. (02 de 08 de 2016). *CRISP-DM: La metodología para poner en orden en los proyectos*. Obtenido de https://www.sngular.com/es/data-science-crisp-dm-metodologia

Rugarcía, A. (1993). La deserción universitaria. *Renglones, revista del ITESO*(26).

Sánchez Guzmán, D., & Rico Páez, A. (01 de 05 de 2018). Análisis de datos de estudiantes de ingeniería paraq la predicción del rendimiento académico mediante técnica de clasificación bayesiana. *Revista Dilemas Contemporáneos: Educación, Política y Valores*(3), 1-23.

Saucedo, M., Herrera-Sánchez, S., Díaz, J., & Bautista, S. y. (2014). Indicadores de reprobación: Facultad de Ciencias Educativas (UNACAR). *Revista Iberoamericana para la Investigación y el Desarrollo Educativo RIDE, 5(9)*, 1-11.

Sucar, L. E. (2006). Redes bayesianas. Aprendizaje Automático: conceptos básicos y avanzados. 77-100.

Taskesen, E. (2019). *bnlearn*. Obtenido de https://github.com/erdogant/bnlearn

Taskesen, E. (2019). *bnlearn*. Obtenido de https://github.com/erdogant/bnlearn

team, T. p. (02 de 2020). *Pandas*. Obtenido de https://pandas.pydata.org/

Vergara, G., & Peredo, H. (2017). Relación del desempeño académico de estudiantes de primer año de universidad en Chile y los instrumentos de selección para su ingreso. *Revista Educación, 41*(2), 1-16.

Waskom, M. (2021). seaborn: statistical data visualization. (J. o. Software, Ed.) *6*(60), 3021. doi:10.21105/joss.03021

9 ANEXOS

9.1 Trama de experimentos del modelo implementado

Se presentan los DAG de los 5 experimentos realizados con el modelo bayesiano implementado, donde se puede observar el grafo acíclico dirigido obtenido una vez aprendida la estructura de los datos, tanto de la porción de entrenamiento como de la de pruebas.

Posteriormente, una vez que se a aprendido como es la estructura, se obtiene el DAG con la distribución de probabilidad condicional (CPD) del modelo, obteniendo como resultado los grafos acíclicos dirigidos de la porción de entrenamiento y de pruebas de cada experimento.

9.1.1 Experimento 1 (FOLD 1):

9.1.1.1 Aprendizaje de Estructura (TRAIN/FOLD 1)

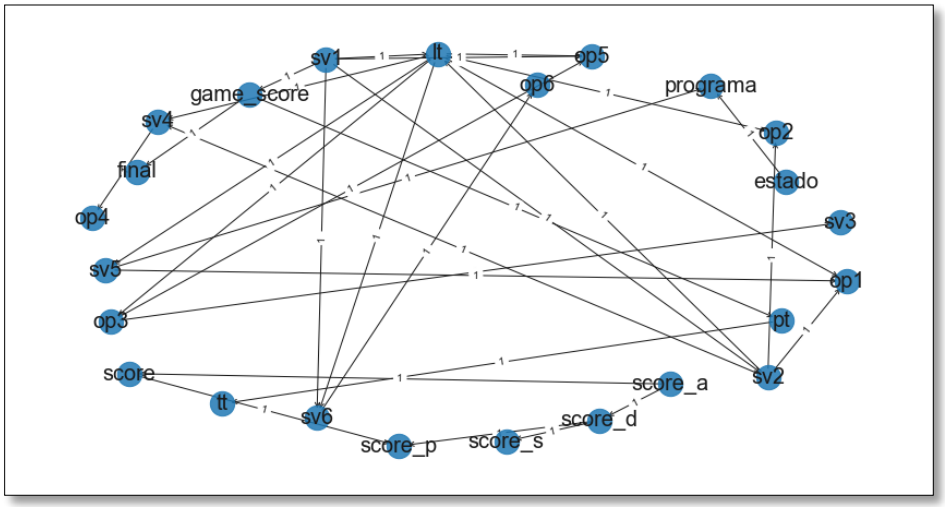

Figura 24: Structure Learning - Train - Fold 1

9.1.1.2 Aprendizaje de Parámetros (TRAIN/FOLD 1)

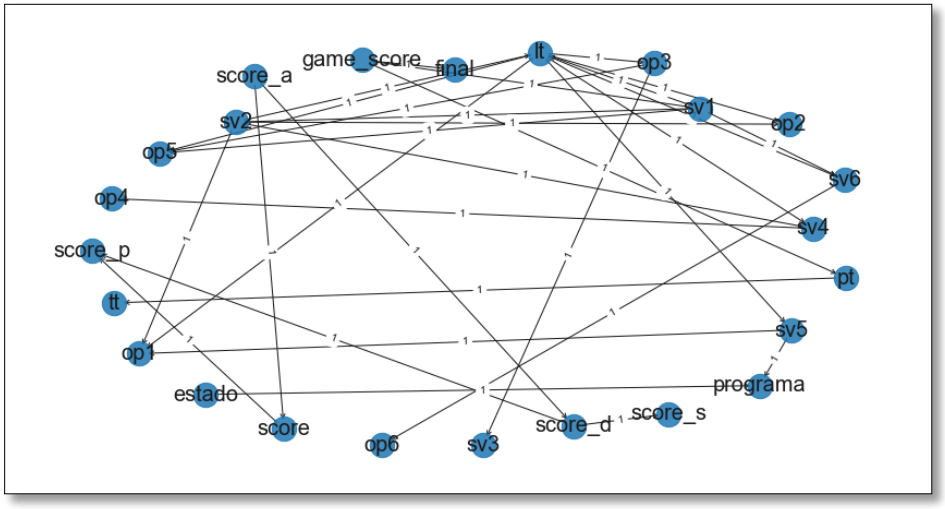

Figura 25: Parameter Learning - Train - Fold 1

9.1.1.3 Aprendizaje de Estructura (TEST/FOLD 1)

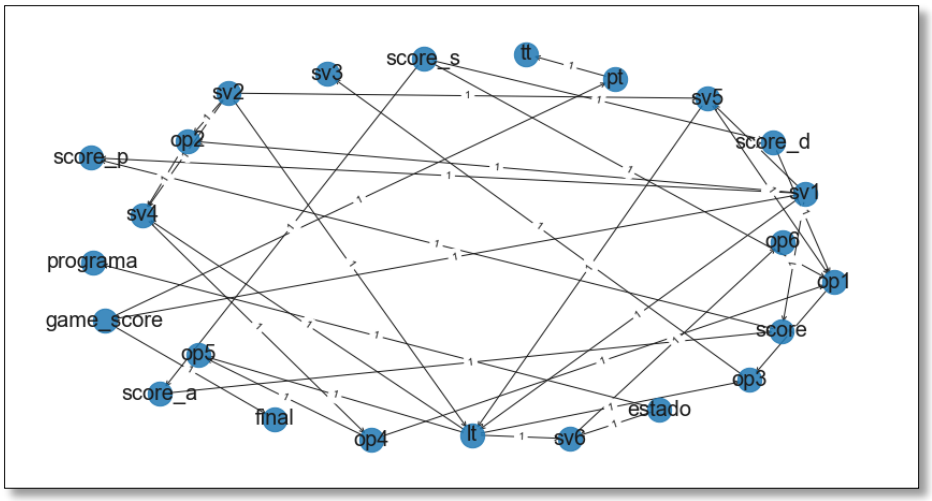

Figura 26: Structure Learning - Test - Fold 1

9.1.1.4 Aprendizaje de Parámetros (TEST/FOLD 1)

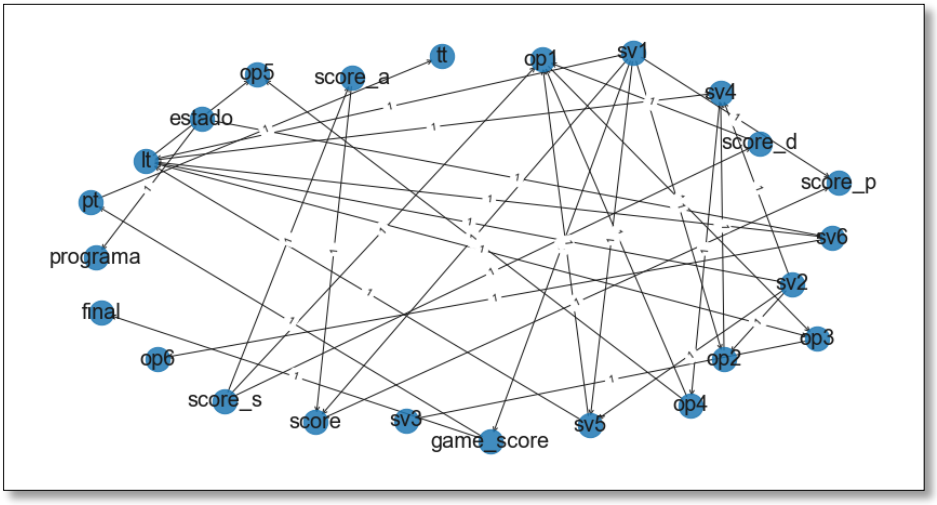

Figura 27: Parameter Learning - Test - Fold 1

9.1.2 Experimento 2 (FOLD 2):

9.1.2.1 Aprendizaje de Estructura (TRAIN/FOLD 2)

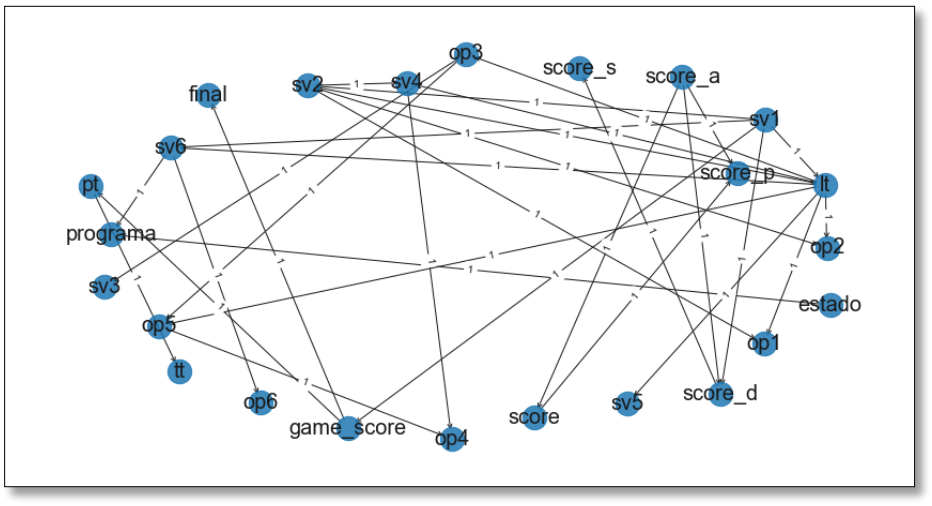

Figura 28: Structure Learning - Train - Fold 2

9.1.2.2 Aprendizaje de Parámetros (TRAIN/FOLD 2)

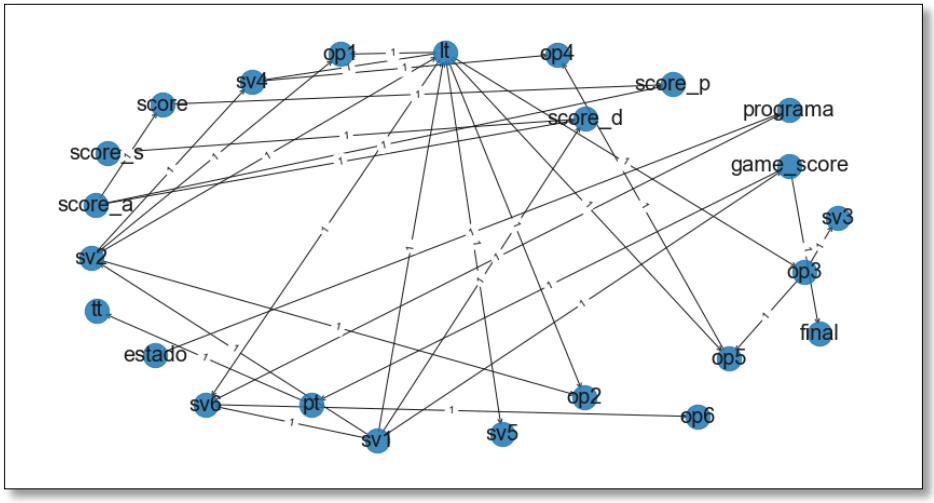

Figura 29: Parameter Learning - Train - Fold 2

9.1.2.3 Aprendizaje de Estructura (TEST/FOLD 2)

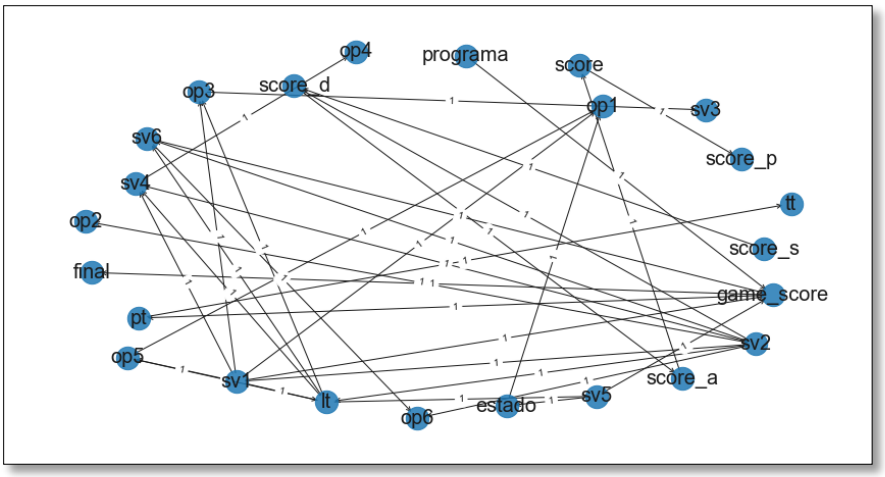

Figura 30: Structure Learning - Test - Fold 2

9.1.2.4 Aprendizaje de Parámetros (TEST/FOLD 2)

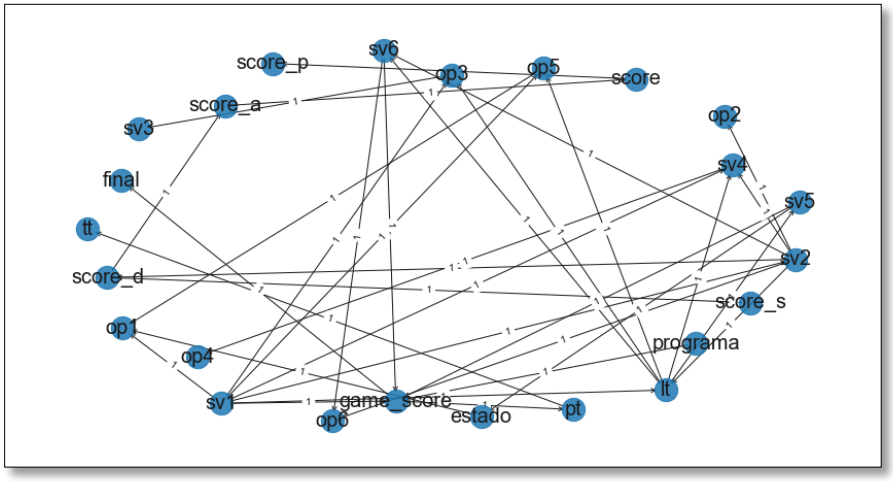

Figura 31: Parameter Learning - Test - Fold 2

9.1.3 Experimento 3 (FOLD 3):

9.1.3.1 Aprendizaje de Estructura (TRAIN/FOLD 3)

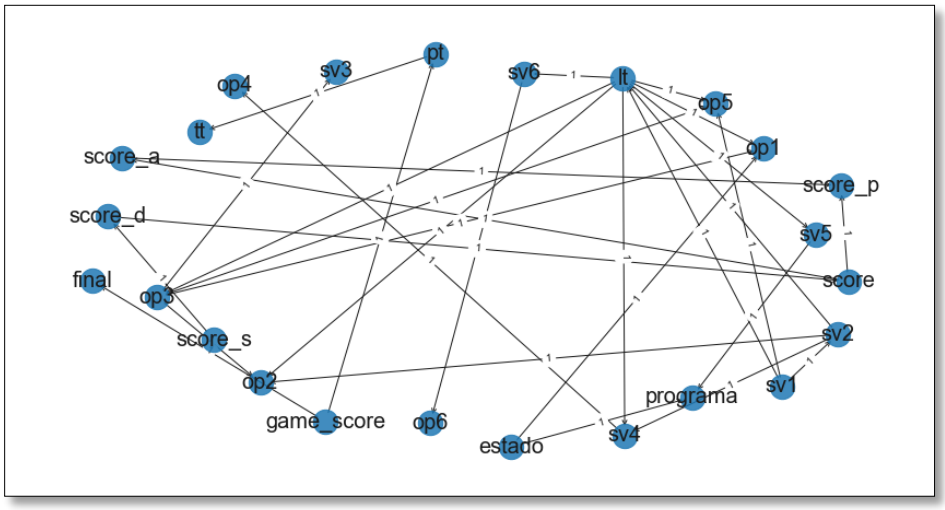

Figura 32: Structure Learning - Train - Fold 3

9.1.3.2 Aprendizaje de Parámetros (TRAIN/FOLD 3)

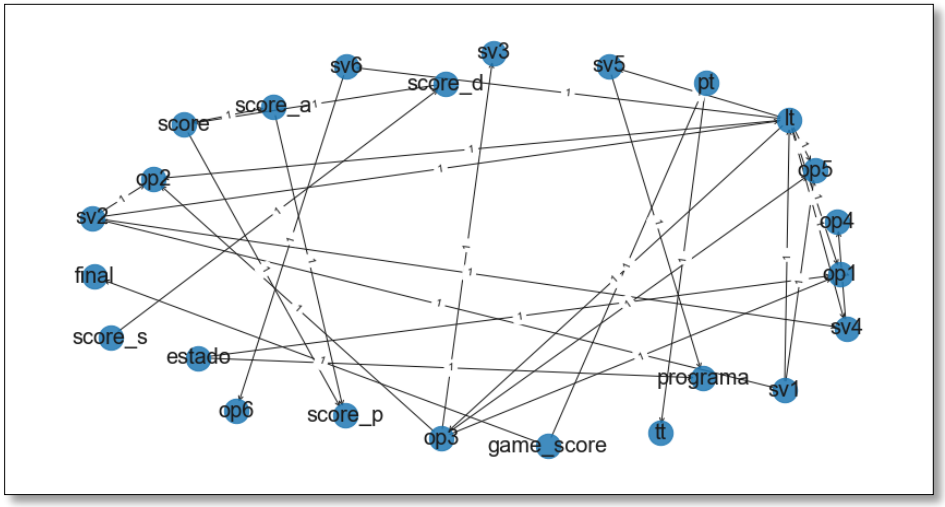

Figura 33: Parameter Learning - Train - Fold 3

9.1.3.3 Aprendizaje de Estructura (TEST/FOLD 3)

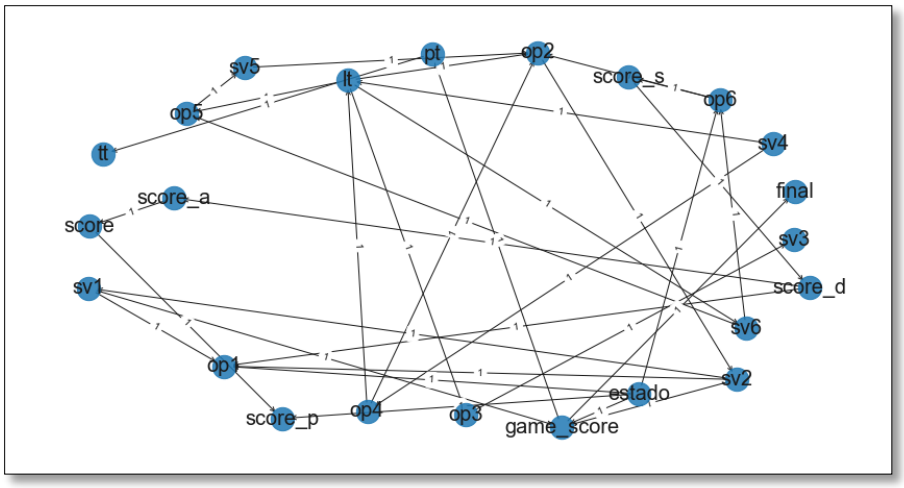

Figura 34: Structure Learning - Test - Fold 3

9.1.3.4 Aprendizaje de Parámetros (TEST/FOLD 3)

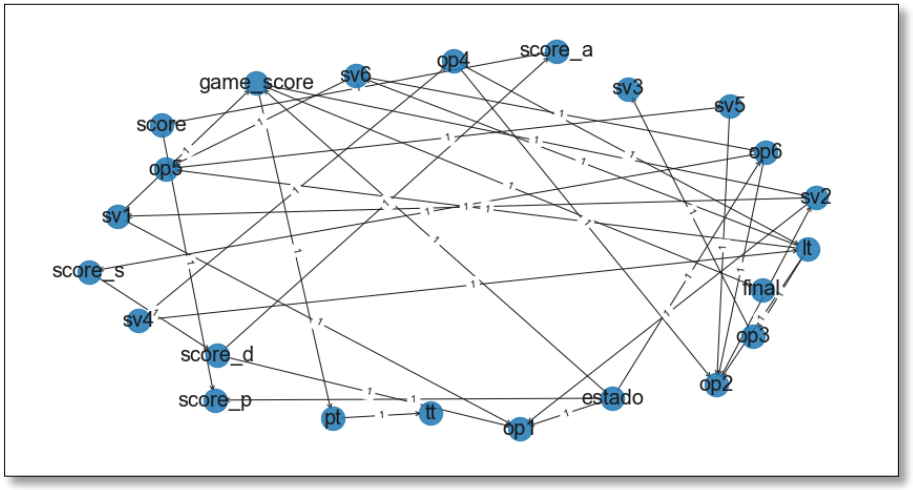

Figura 35: Parameter Learning - Test - Fold 3

9.1.4 Experimento 4 (FOLD 4):

9.1.4.1 Aprendizaje de Estructura (TRAIN/FOLD 4)

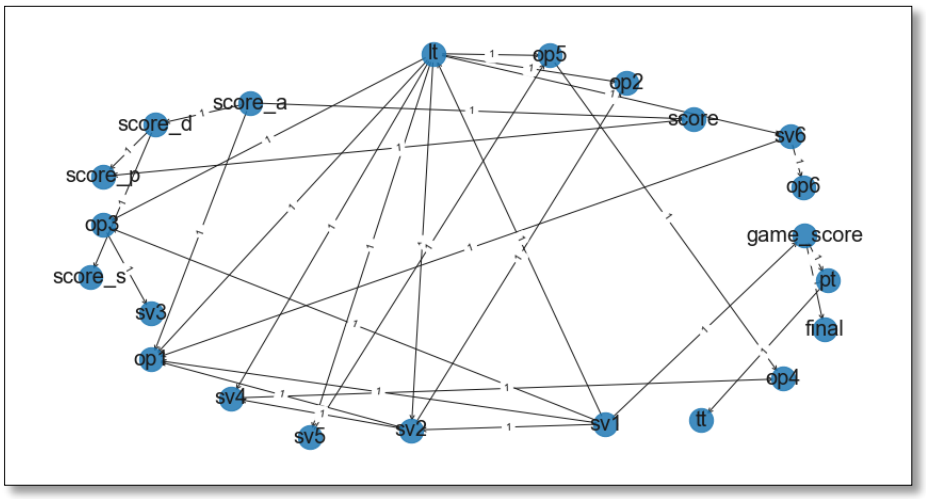

Figura 36: Structure Learning - Train - Fold 4

9.1.4.2 Aprendizaje de Parámetros (TRAIN/FOLD 4)

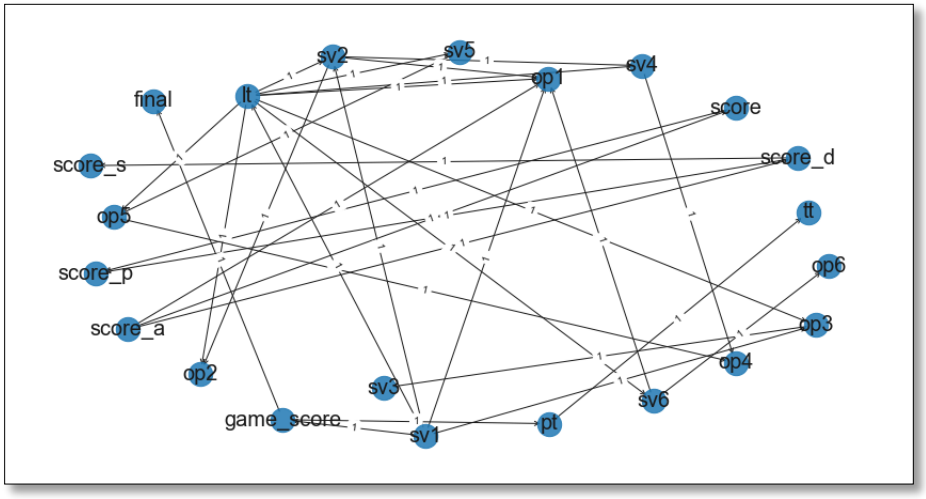

Figura 37: Parameter Learning - Train - Fold 4

9.1.4.3 Aprendizaje de Estructura (TEST/FOLD 4)

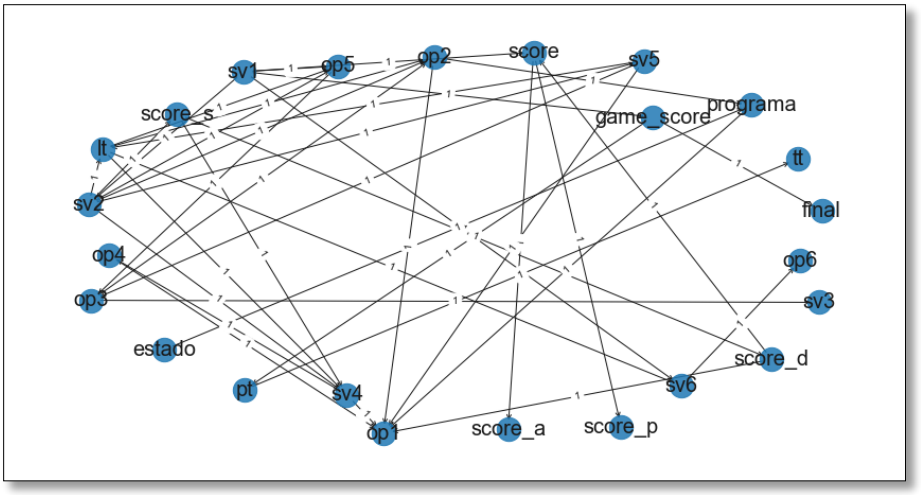

Figura 38: Structure Learning - Test - Fold 4

9.1.4.4 Aprendizaje de Parámetros (TEST/FOLD 4)

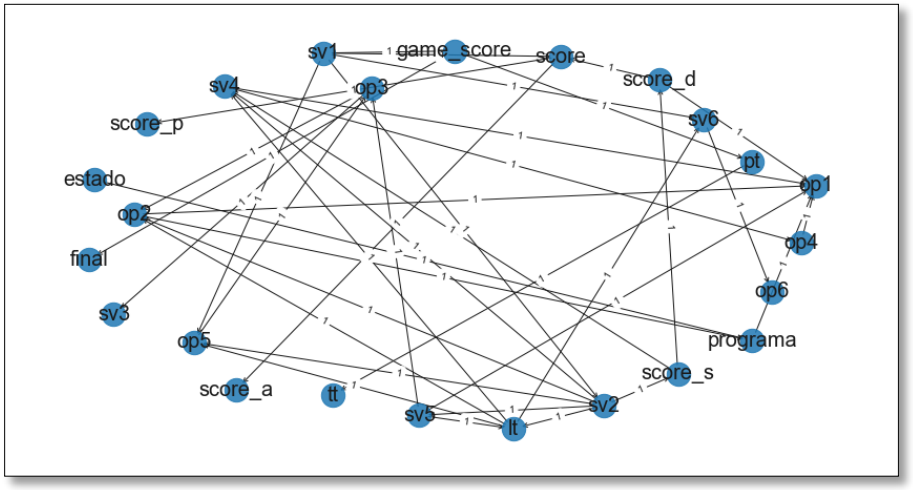

Figura 39: Parameter Learning - Test - Fold 4

9.1.5 Experimento 4 (FOLD 5):

9.1.5.1 Aprendizaje de Estructura (TRAIN/FOLD 5)

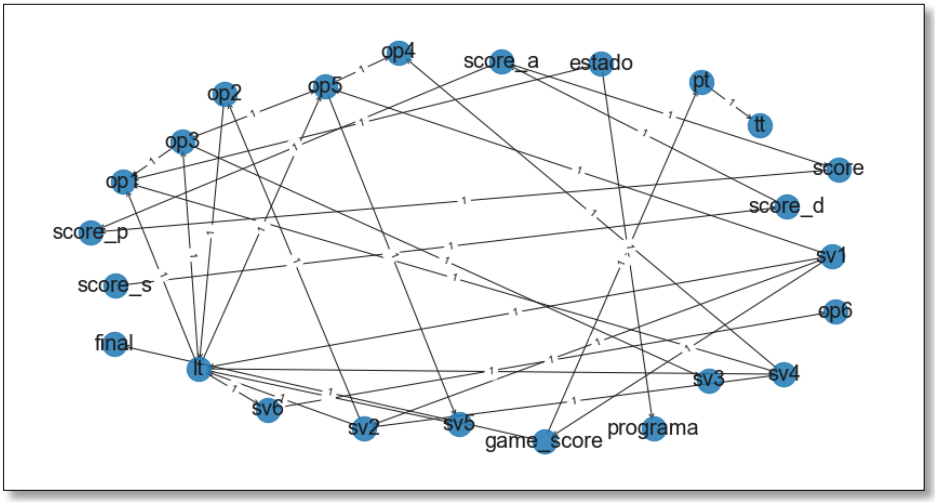

Figura 40: Structure Learning - Train - Fold 5

9.1.5.2 Aprendizaje de Parámetros (TRAIN/FOLD 5)

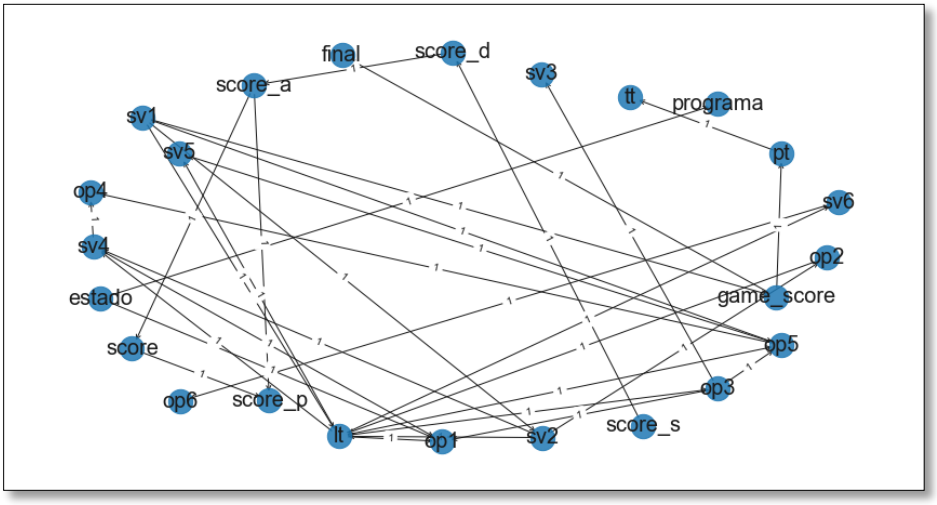

Figura 41: Parameter Learning - Train - Fold 5

9.1.5.3 Aprendizaje de Estructura (TEST/FOLD 5)

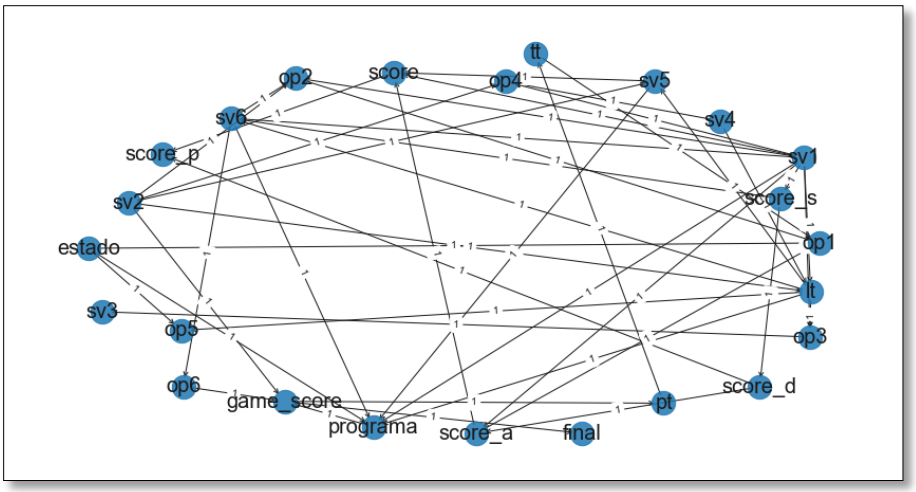

Figura 42: Structure Learning - Test - Fold 5

9.1.5.4 Aprendizaje de Parámetros (TEST/FOLD 5)

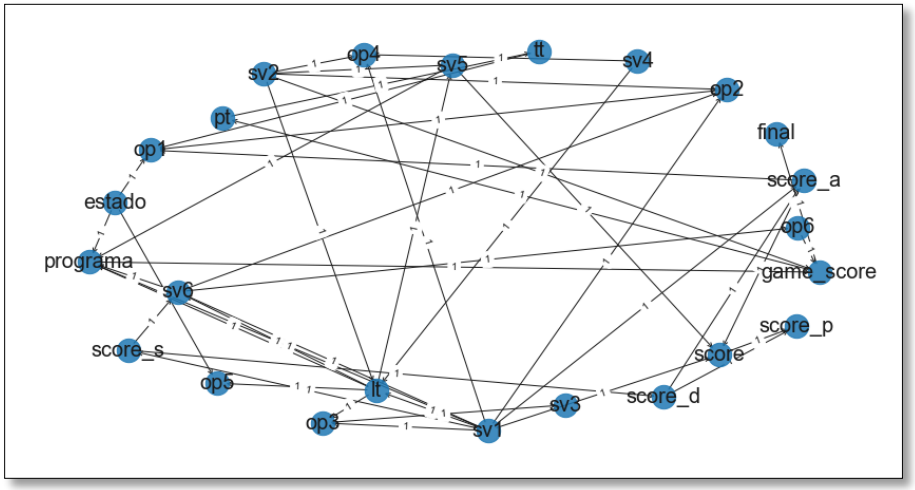

Figura 43: Parameter Learning - Test - Fold 5

9.2 Grafo generado en Weka

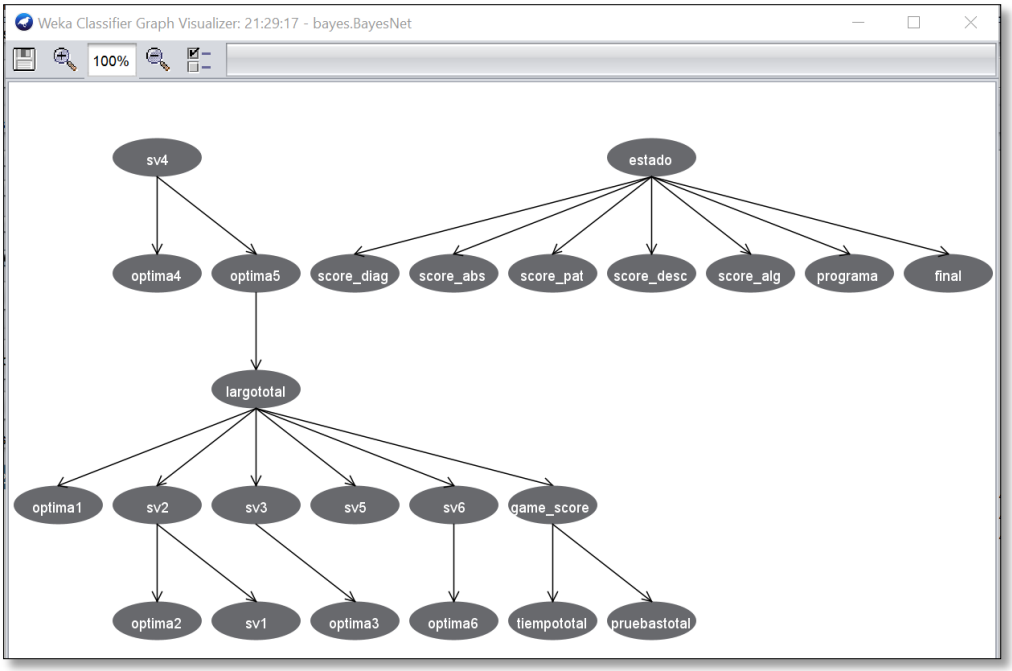

Figura 44: Grafo Acíclico Dirigido - Weka

9.3 Datos de salida en Weka

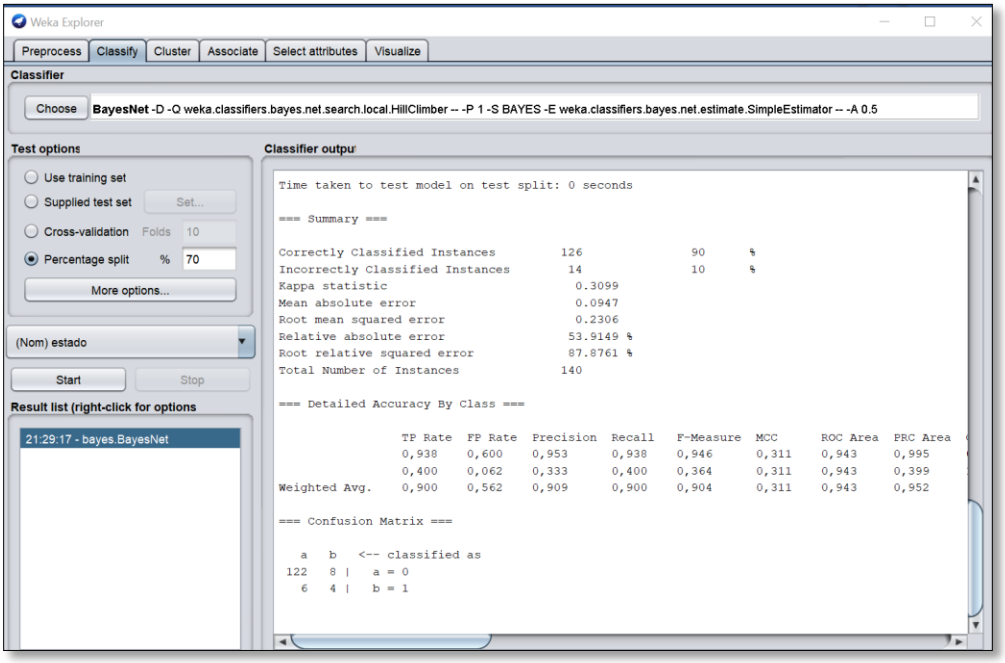

Figura 45: Resultados Experimento Weka

9.4 Tablas de Distribución probabilística en Weka

Distribución probabilística de las variables “estado”, “score_diag”, “programa” y “final”

Figura 46: Distribución de Probabilidad - "estado"

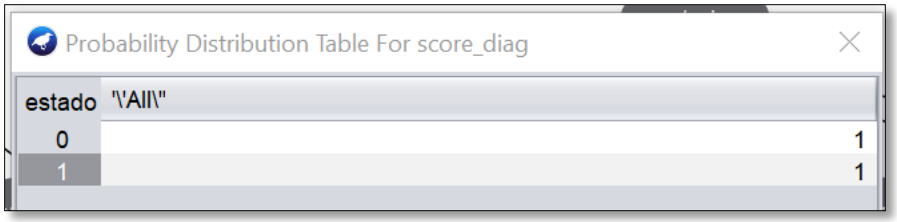

Figura 47: Distribución de Probabilidad "score"

Probability Distribution Table For programa

estado	1	2	3	4	5
0	0,272	0,427	0,058	0,237	0,006
1	0,456	0,184	0,146	0,204	0,01

Figura 48: Distribución de Probabilidad "programa"

Probability Distribution Table For final

estado	'\'(-inf-4.55]\"	'\'(4.55-5.3...	'\'(5.35-5.8...	'\'(5.85-6.3...	'\'(6.35-inf)\"
0	0,094	0,222	0,244	0,22	0,22
1	0,961	0,01	0,01	0,01	0,01

Figura 49: Distribución de Probabilidad "final"