



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

1	INTRODUCCIÓN	17
2	IDENTIFICACIÓN DEL PROBLEMA.....	19
2.1	Obtención de fuentes de datos.....	19
3	OBJETIVOS E HIPÓTESIS	20
3.1	Objetivo General	20
3.2	Objetivos Específicos.....	20
3.3	Hipótesis.....	20
3.4	Preguntas de investigación	20
4	MARCO TEÓRICO.....	22
4.1	Rendimiento Académico asociado a la deserción.....	22
4.2	Redes Bayesianas.....	24

4.2.1	Librería bnlearn	28
4.2.2	Algoritmos de aprendizaje de estructuras	28
4.2.2.1	Algoritmo Hill-Climbing (HC).....	29
4.2.2.2	Aprendizaje basado en medida de puntuación.....	30
4.2.3	Algoritmos de aprendizaje de parámetros	35
4.2.3.1	Estimador de Máxima Verosimilitud (MLE).....	36
4.2.4	Algoritmos de inferencia	37
4.2.4.1	Algoritmo de Eliminación de Variables	38
4.2.5	Funciones de validación de modelos (métricas).....	38
4.2.5.1	Matriz de Confusión.....	39
4.2.5.2	Accuracy Score (Puntuación de Exactitud)	39
4.2.5.3	Balanced Accuracy Score (Puntación de Exactitud Equilibrada)....	40
4.2.5.4	Precision Score	40
4.2.5.5	Recall Score (Puntuación de Recuperación)	41
4.2.5.6	Curva ROC - AUC	41
4.2.6	Método de Selección de Variables	43
4.2.7	Técnica de Sobremuestreo (SMOTE)	45
4.2.7.1	Algoritmo de SMOTE.....	46
4.3	Investigación sobre estudios relacionados a la predicción académica	47
4.3.1	Análisis de datos de estudiantes de ingeniería para la predicción del rendimiento académico mediante técnica de clasificación bayesiana.....	47
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	48
4.3.3	Análisis de datos educativos utilizando Redes Bayesianas	49
4.3.4	Análisis de la progresión de los estudiantes en una asignatura introductoria a la programación mediante Redes Bayesianas	51
4.3.5	Análisis del alumnado de la Universidad de Almería mediante Redes Bayesianas.....	52
4.4	Metodologías de ciencia de datos.....	54

4.4.1 Proceso KDD.....	54
4.4.2 Metodología CRISP-DM	56
4.4.3 Metodología SEMMA.....	58
4.5 Algoritmos y Técnicas para la minería de datos.....	60
4.6 Metodología de gestión de proyectos.....	62
4.6.1 PMBOK.....	62
4.6.2 SCRUM	62
5 METODOLOGÍA DE TRABAJO	64
5.1 Enfoque de la investigación.....	64
5.2 Metodología de análisis de datos.....	64
5.3 Obtención de datos.....	66
5.4 Metodología de gestión de proyectos.....	66
5.4.1 Hitos del proyecto.....	66
5.4.2 Diagrama de Gantt	67
6 APLICACIÓN METODOLÓGICA	68
6.1 Resumen.....	68
6.2 Análisis de Datos	69
6.2.1 Análisis 1D.....	73
6.2.2 Análisis 2D.....	77
6.3 Preprocesamiento de datos	80
6.4 Experimentos	82
6.4.1 Diseño de experimentos	82
6.4.2 Modelo bajo Python	85
6.4.2.1 Resultados de experimentos de modelo bajo Python.....	86
6.4.2.2 Análisis de red Bayesiana (modelos Python)	87
6.4.3 Modelo bajo R	90
6.4.3.1 Resultado de experimentos de modelo bajo R	91
6.4.3.2 Análisis de redes Bayesianas (modelo R)	103
6.4.4 Resultados de modelos bajo Weka	108
6.5 Requerimientos Técnicos	112

6.5.1	Requerimientos para la ejecución de los modelos	112
6.5.1.1	Requerimientos de Hardware.....	112
6.5.1.2	Requerimientos de Software	112
6.5.2	Configuración de ambiente.....	112
6.5.3	Librerías utilizadas.....	113
7	CONCLUSIONES	115
7.1	Resultados del Modelo Bayesiano	116
7.2	Resultados del Modelo Weka.....	117
8	REFERENCIAS BIBLIOGRÁFICAS.....	120
9	ANEXOS	125
9.1	DAGs de los experimentos ejecutados con R sobre Python	125

9.1.1 Experimento con datos balanceados y discretos con medida de puntuación AIC	125
9.1.2 Experimento con datos balanceados y discretos con medida de puntuación BIC	126
9.1.3 Experimento con datos balanceados, discretos y continuos con medida de puntuación AIC-CG	127
9.1.4 Experimento con datos balanceados, discretos y continuos con medida de puntuación AIC-CG con selección de 5 variables	128
9.1.5 Experimento con datos balanceados, discretos y continuos con medida de puntuación AIC-CG con selección de 10 variables	129
9.1.6 Experimento con datos balanceados, discretos y continuos con medida de puntuación AIC-CG con selección de 15 variables	130
9.1.7 Experimento con datos balanceados, discretos y continuos con medida de puntuación AIC-CG con selección de 20 variables	131
9.1.8 Experimento con datos balanceados, discretos y continuos con medida de puntuación AIC-CG con aplicación de restricción de arcos	132
9.1.9 Experimento con datos balanceados, discretos y continuos con medida de puntuación BIC-CG	133
9.1.10 Experimento con datos desbalanceados y discretos con medida de puntuación AIC.....	134
9.1.11 Experimento con datos desbalanceados y discretos con medida de puntuación AIC con selección de 5 variables.....	135
9.1.12 Experimento con datos desbalanceados y discretos con medida de puntuación AIC con selección de 10 variables.....	136
9.1.13 Experimento con datos desbalanceados y discretos con medida de puntuación AIC con selección de 15 variables.....	137
9.1.14 Experimento con datos desbalanceados y discretos con medida de puntuación AIC con selección de 20 variables.....	138
9.1.15 Experimento con datos desbalanceados y discretos con medida de puntuación AIC con aplicación de restricción de arcos	139

9.1.16 Experimento con datos desbalanceados y discretos con medida de puntuación BIC.....	140
9.1.17 Experimento con datos desbalanceados, discretos y continuos con medida de puntuación AIC-CG	141
9.1.18 Experimento con datos desbalanceados, discretos y continuos con medida de puntuación BIC-CG	142
9.1.19 Experimento con datos desbalanceados, discretos y continuos con medida de puntuación LOGLIK-CG	143
9.2 DAGs de los experimentos ejecutados con Python	144
9.2.1 Experimento con datos balanceados con medida de puntuación BDEU ..	144
9.2.2 Experimento con datos balanceados con medida de puntuación K2	145
9.2.3 Experimento con datos balanceados con medida de puntuación BIC	146
9.2.4 Experimento con datos desbalanceados con medida de puntuación BDEU	
147	
9.2.5 Experimento con datos desbalanceados con medida de puntuación K2..	148
9.2.6 Experimento con datos desbalanceados con medida de puntuación BIC	149
9.3 Grafos generados en Weka.....	150
9.3.1 Experimento con variables del juego diagnóstico y algoritmo K2 Global..	150
9.3.2 Experimento con variables del juego diagnóstico y algoritmo TabuSearch Global	150
9.3.3 Experimento con variables del juego diagnóstico y algoritmo Repeated Hill Climber Global	151
9.3.4 Experimento con variables del juego diagnóstico y algoritmo Hill Climber Local	152
9.3.5 Experimento con variables del juego diagnóstico y algoritmo Repeated Hill Climber Local	153
9.3.6 Experimento con variables del juego diagnóstico y algoritmo Hill Climber Global desbalanceado	154

INDICE DE TABLAS

Tabla 1: Matriz de Confusión Binaria.....	39
Tabla 2: Relación entre las áreas de conocimiento y los macroprocesos.	62
Tabla 3: Resumen comparativo entre KDD, SEMMA y CRISP-DM (Azevedo & Santos, 2008).	64
Tabla 4: Hitos del proyecto.....	67
Tabla 5: Diagrama de Gantt.	67
Tabla 6: Lista de atributos descartados del análisis	70
Tabla 7: Variables relacionadas al juego de diagnóstico.....	72
Tabla 8: Variables relacionadas con la prueba de diagnóstico.....	72
Tabla 9: Variables relacionadas al curso de programación.....	73
Tabla 10: Conversión de valores en variable “programa”.....	73
Tabla 11: Conversión de valores en variable "estado".	73
Tabla 12: Tabla de contingencia programa / estado.....	79
Tabla 13: Tabla de contingencia op1 / sv1.....	79
Tabla 14: Tabla de contingencia op2 / sv2.....	79
Tabla 15: Tabla de contingencia op3 / sv3.....	79
Tabla 16: Tabla de contingencia op4 / sv4.....	80
Tabla 17: Tabla de contingencia op5 / sv5.....	80
Tabla 18: Tabla de contingencia op6 / sv6.....	80
Tabla 19: Listado de variables seleccionadas para realizar el modelo.....	81
Tabla 20: Lista de variables discretizadas.....	81
Tabla 21: Resultados set de validación desbalanceado (modelos bajo Python).	86
Tabla 22: Resultados set de validación balanceado (modelos bajo Python).....	86
Tabla 23: Particiones para BDEU con datos balanceados con foco en la métrica Accuracy.	87
Tabla 24: Listado de experimentos ejecutados utilizados ejecutando R sobre Python, 90	90
Tabla 25: Resultado de métricas para el set de validación para 9 experimentos ejecutados.	91
Tabla 26: Listado de experimentos ejecutados por método de selección de variables. 92	92

Tabla 27: Métricas de los experimentos seleccionados para la aplicación del método de selección de variables.	92
Tabla 28: Tabla comparativa de experimento con datos desbalanceados y discretos con medida de puntuación AIC con respecto a la aplicación del método selección de variables (5, 10, 15 y 20 variables).	93
Tabla 29: Tabla comparativa de experimento con datos balanceados, discretos y continuos con medida de puntuación AIC-CG con respecto a la aplicación del método de selección de variables (5, 10 ,15, 20 variables).....	93
Tabla 30: Tabla de métricas por partición del experimento con datos desbalanceados y discretos usando la medida de puntuación AIC.....	94
Tabla 31: Tabla de métricas por partición del experimento con datos balanceados, discretos y continuos con la medida de puntuación AIC-CG.....	94
Tabla 32: Resultado de métricas de los experimentos seleccionados una vez aplicada las restricciones de arcos.....	101
Tabla 33: Tabla comparativa de experimento con datos balanceados, discretos y continuos usando medida de puntuación AIC-CG con respecto a la aplicación de restricción de arcos.	101
Tabla 34: Tabla comparativa de experimento con datos desbalanceados y discretos usando medida de puntuación AIC con respecto a la aplicación de restricción de arcos.	101
Tabla 35: Tabla comparativa de experimento con datos desbalanceados y discretos usando medida de puntuación AIC por cada partición con respecto a la aplicación de restricción de arcos.	102
Tabla 36: Comparativa de experimentos con datos desbalanceados y discretos usando medida de puntuación AIC.	102
Tabla 37: Comparativa de experimentos con datos balanceados, discretos y continuos usando medida de puntuación AIC-CG.	103
Tabla 38: Particiones del experimento con datos desbalanceados y discretos con medida de puntuación AIC con método de selección de 5 variables.	104
Tabla 39: Particiones del experimento con datos balanceados, discretos y continuos con medida de puntuación AIC-CG con restricción de arcos.	104

Tabla 40: Comparación de métricas de los dos algoritmos con mejor desempeño.....	109
Tabla 41: Comparación de métricas de modelo seleccionado desbalanceado y balanceado.....	109
Tabla 42: Comparación de métricas de algoritmo Hill Climber Global para dos estimadores de probabilidades.....	110
Tabla 43: Comparación de métricas de modelo seleccionado desbalanceado y balanceado.....	111

INDICE DE FIGURAS

Figura 1: Formula de Bayes, también conocida como Regla de Bayes.	18
Figura 2: Ejemplo de una relación de influencia causal.	27
Figura 3: Ejemplo de una red Bayesiana.	27
Figura 4: Algoritmo de Hill-Climbing.	30
Figura 5: Curva ROC.	42
Figura 6: Área bajo la curva (AUC).	43
Figura 7: Proceso KDD (Fayyad, Piatetsky-Shapiro, & Smyth, 1996).	55
Figura 8: Metodología CRISP-DM.	58
Figura 9: Metodología SEMMA.	59
Figura 10: Proceso de Scrum.	63
Figura 11: Encuesta KDnuggets sobre métodos análisis, minería de datos o ciencia de datos.	65
Figura 12: Lectura de la base de datos.	74
Figura 13: Verificación de valores nulos.	74
Figura 14: Análisis 1D de las primeras 15 variables.	74
Figura 15: Análisis 1D de las siguientes 14 variables.	75
Figura 16: Análisis 1D de las siguientes 14 variables.	75
Figura 17: Análisis 1D de las últimas 14 variables.	76
Figura 18: Cálculo de porcentaje para las variables "programa" y "estado".	76
Figura 19: Matriz de correlación de variables.	78
Figura 20: DAG - Bdeu - Balanceado - Partición 2.	88
Figura 21: Red con datos desbalanceados y discreto usando medida de puntuación AIC.	95
Figura 22: Red con datos desbalanceados y discreto usando medida de puntuación AIC después de aplicar restricciones.	97
Figura 23: Red con datos balanceados, discretos y continuos usando medida de puntuación AIC-CG.	98
Figura 24: Red con datos balanceados, discretos y continuos usando medida de puntuación AIC-CG después de aplicar restricciones.	100

Figura 25: Red con datos desbalanceados y discretos con medida de puntuación AIC y con método de selección de 5 variables.....	105
Figura 26: Red con datos balanceados, discretos y continuos con medida de puntuación AIC-CG y con restricción de arcos	106
Figura 27: Grafo generado con el algoritmo seleccionado Hill Climber Global balanceado.	110
Figura 28: Grafo generado con el modelo seleccionado balanceado.....	111
Figura 29: Red de la partición 1 del experimento con datos balanceados y discretos con medida de puntuación AIC.	125
Figura 30: Red de la partición 1 del experimento con datos balanceados y discretos con medida de puntuación BIC.	126
Figura 31: Red de la partición 1 del experimento con datos balanceados, discretos y continuos con medida de puntuación AIC-CG.....	127
Figura 32: Red de la partición 1 del experimento con datos balanceados, discretos y continuos con medida de puntuación AIC-CG con selección de 5 variables.....	128
Figura 33: Red de la partición 1 del experimento con datos balanceados, discretos y continuos con medida de puntuación AIC-CG con selección de 10 variables.....	129
Figura 34: Red de la partición 1 del experimento con datos balanceados, discretos y continuos con medida de puntuación AIC-CG con selección de 15 variables.....	130
Figura 35: Red de la partición 1 del experimento con datos balanceados, discretos y continuos con medida de puntuación AIC-CG con selección de 20 variables.....	131
Figura 36: Red de la partición 1 del experimento con datos balanceados, discretos y continuos con medida de puntuación AIC-CG con aplicación de restricción de arcos.	132
Figura 37: Red de la partición 1 del experimento con datos balanceados, discretos y continuos con medida de puntuación BIC-CG.....	133
Figura 38: Red de la partición 1 del experimento con datos desbalanceados y discretos con medida de puntuación AIC.....	134
Figura 39: Red de la partición 2 del experimento con datos desbalanceados y discretos con medida de puntuación AIC con selección de 5 variables.....	135
Figura 40: Red de la partición 1 de experimento con datos desbalanceados y discretos con medida de puntuación AIC con selección de 10 variables.....	136

Figura 41: Red de la partición 1 del experimento con datos desbalanceados y discretos con medida de puntuación AIC con selección de 15 variables.....	137
Figura 42: Red de la partición 1 del experimento con datos desbalanceados y discretos con medida de puntuación AIC con selección de 20 variables.....	138
Figura 43: Red de la partición 1 del experimento con datos desbalanceados y discretos con medida de puntuación AIC con aplicación de restricción de arcos.....	139
Figura 44: Red de la partición 1 del experimento con datos desbalanceados y discretos con medida de puntuación BIC.....	140
Figura 45: Red de la partición 1 del experimento con datos desbalanceados, discretos y continuos con medida de puntuación AIC-CG.....	141
Figura 46: Red de la partición 1 del experimento con datos desbalanceados, discretos y continuos con medida de puntuación BIC-CG.....	142
Figura 47: Red de la partición 1 del experimento con datos desbalanceados, discretos y continuos con medida de puntuación LOGLIK-CG.....	143
Figura 48: Red de la partición 1 del experimento con datos balanceados con medida de puntuación BDEU.....	144
Figura 49: Red de la partición 1 del experimento con datos balanceados con medida de puntuación K2.....	145
Figura 50: Red de la partición 1 del experimento con datos balanceados con medida de puntuación BIC.....	146
Figura 51: Red de la partición 1 del experimento con datos desbalanceados con medida de puntuación BDEU.....	147
Figura 52: Red de la partición 1 del experimento con datos desbalanceados con medida de puntuación K2.....	148
Figura 53: Red de la partición 1 del experimento con datos desbalanceados con medida de puntuación BIC.....	149
Figura 54: Red de primer experimento con datos desbalanceados y estimador de probabilidades MLE.....	150
Figura 55: Red de primer experimento con datos desbalanceados y estimador de probabilidades MLE.....	150

Figura 56: Red de primer experimento con datos desbalanceados y estimador de probabilidades MLE.....	151
Figura 57: Red de primer experimento con datos desbalanceados, métrica AIC y estimador de probabilidades simple alfa 0.5.....	152
Figura 58: Red de primer experimento con datos desbalanceados, métrica AIC y estimador de probabilidades MLE	153
Figura 59: Red de primer experimento con datos desbalanceados y estimador de probabilidades MLE.....	154

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. Más 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 reproban, 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. La información obtenida corresponde al año 2019 de alumnos de primer año de carreras de Ingeniería de una universidad chilena.

En cuanto a los resultados obtenidos, estos muestran un buen desempeño del modelo aplicado donde se obtuvieron métricas por sobre el 60% en la mayoría de los experimentos y llegando a resultados óptimo por sobre el 80% lo que concluyen que el método propuesto es viable de utilizar para diagnosticar las posibles reprobaciones de los alumnos.

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. Furthermore, there are few studies that consider the assessment 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 face this problem, the use of Bayesian networks is chosen, taking a data sample to estimate the probability of the variables and their dependencies, allowing the identification of 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 previously carried out in a programming course. The information obtained corresponding to the year 2019 of first-year engineering students from a Chilean university.

Regarding the results obtained, these showed a good performance of the applied model where metrics were obtained above 60% in most of the experiments and reaching optimal results above 80%, which concludes that the proposed method is viable of use to diagnose possible student failures.

1 INTRODUCCIÓN

Se han realizado una diversidad de estudios relacionados al rendimiento académico y de deserción en universidades chilenas, de estos se ha 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.

Especificamente, en cuanto a la investigación de aptitudes intelectuales, el conjunto de universidades, ubicadas en distintos países, tanto de América como de otros continentes, es bastante amplio y heterogéneo. Como no es recomendable 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 reproban 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 reproban 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 niveleen 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).

Considerando la deserción a nivel mundial, la OECD ha analizado los datos de 18 países, concluyendo que el 30% de los estudiantes que ingresan a la universidad no se gradúan de la educación superior. En países como México, Nueva Zelanda, Suecia, y los Estados Unidos, el porcentaje es mayor a 40%, mientras que, en otros países como Francia, Japón, Corea, España y Rusia, es menor a 25%. En Sudáfrica, cerca del 50% de los estudiantes inscritos en educación superior desertaron en los primeros 3 años (UNESCO-IESALC, 2020).

En el libro “Student Affairs and Services in Higher Education: Global Foundations, Issues, and Best Practices”, se mencionan las siguientes tasas de retención: Australia (67%), Irlanda (83%), Países Bajos (76%), Reino Unido (78%) y Estados Unidos (64%). En Marruecos, 58% de los estudiantes abandonan antes de la graduación. En Latinoamérica las tasas de deserción son altas, con 73% de estudiantes que no se gradúan de universidades públicas y privadas. Otras tasas de deserción que se mencionan son 50%

para Brasil, 41% para Chile y 39% para México. En Asia estas tasas varían entre países, y en África son altas (Ludeman & Schreiber, 2020).

En el contexto de Chile, en las últimas décadas las políticas públicas en educación superior han cambiado su enfoque de incrementar la cantidad de estudiantes a la calidad de la educación. Por lo tanto, los indicadores que dan cuenta del proceso formativo y sus resultados, como es la deserción de primer año, han cobrado gran relevancia para las instituciones y el Estado. La importancia de estimar la deserción también es debida a los estudios realizados, que han mostrado el alcance de los costos en los cuales incurre este fenómeno, siendo de índole tanto económica como psicosocial. En relación con los postulantes a la educación superior, los esfuerzos son realizados por las instituciones y también por diversos organismos para proveerles de información acerca de la oferta académica, acreditaciones y otros aspectos, con el fin de ayudarles a evaluar las opciones y hacer una elección de carrera acorde a la vocación, de esta manera se contribuye a reducir el riesgo de deserción. Respecto a los estudiantes ya matriculados, existen iniciativas institucionales y estatales para reducir la deserción por motivos de rendimiento académico, fundamentalmente sobre la base de programas de nivelación (SIES, 2014).

Cada año, el Servicio de Información de Educación Superior (SIES), perteneciente al Ministerio de Educación, publica un informe de retención de estudiantes de primer año en carreras de pregrado. El informe más reciente indica que la tasa de Retención de estudiantes en Educación Superior, en especial la de primer año, es uno de los indicadores más utilizados a nivel internacional para evaluar la eficiencia interna de las casas de estudio, considerando que la mayor deserción de estudiantes se da en ese período. En cuanto a los resultados, este informe indica que para la cohorte 2019 de universidades, es decir, sin incluir Centros de Formación Técnica e Institutos Profesionales, existe una tasa de retención de 79,7%. También se comparan factores como el tipo de carrera (profesional o técnica), jornada (diurna, vespertina y otras), estado de acreditación de la institución y las carreras como tal. Respecto a estas últimas, considerando el área de ingeniería, Ingeniería Civil Industrial posee una alta retención de

83,9%. Por el contrario, las carreras asociadas a Ingeniería en Computación e Informática muestran una tasa de 67,6%, bastante baja comparada a la anterior (SIES, 2020)

Específicamente, en la Universidad Andrés Bello, para el año 2018 se presentó en promedio una tasa de retención de primer año de 79.8%, además la tasa de aprobación de cursos ese mismo año fue de 86.9%, y considerando sólo los programas de pregrado, la tasa de retención de primer año fue de 80.5%. Aunque las cifras mencionadas son bastante buenas, por el contrario, la tasa de retención de tercer año en pregrado, entre los años 2015 y 2018 disminuyó de 63.2% a 60.5%. En comparación con el promedio de las universidades chilenas, la Universidad Andrés Bello presenta tasas ligeramente superiores. En el año 2017, su tasa de retención de primer año en pregrado sin considerar Bachillerato fue de 79.8%, y en programas diurnos sin Bachillerato fue de 82.7%. En cambio, la tasa promedio de universidades chilenas fue de 78.7% (UNAB, 2019).

Dada la evidencia presentada, la cual justifica la búsqueda por parte de la Universidad Andrés Bello, de formas de analizar y predecir el desempeño académico de estudiantes de primer año de carrera, se lleva a cabo esta investigación basada en redes Bayesianas, con la intención de inferir si existen problemáticas a nivel de aptitudes, que impidan la aprobación del curso de programación de primer año.

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 $\Omega = \{e_1, e_2, \dots, e_n\}$ que es un conjunto y los resultados son los elementos del conjunto. Ejemplo:

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

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:

$$P(E) = P(\{e_{i_1}\}) + P(\{e_{i_2}\}) + \dots + P(\{e_{i_k}\}).$$

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, \dots, E_n tal que $P(E_i) \neq 0$ para todos los i , tenemos que $1 \leq i \leq n$,

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

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

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

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).

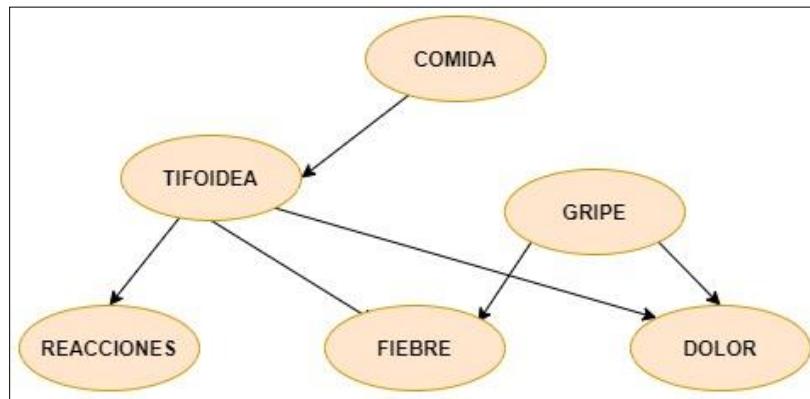


Figura 3: Ejemplo de una red Bayesiana.

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).

4.2.1 Librería bnlearn

bnlearn es un paquete de programación de modelos probabilísticos para *R* y *Python* que permite el aprendizaje de estructura de redes Bayesianas, estimación de sus parámetros, inferencias y métodos de muestreo (Taskesen, 2019). En *Python* este paquete se basa sobre el paquete *pgmpy* que es una biblioteca para trabajar con modelos gráficos probabilísticos. (Ankan, 2015). En *R* este paquete, que fue lanzado en el año 2007, está en continuo desarrollo y provee un amplio abanico de algoritmos de aprendizaje basado en restricciones, puntuación e híbridos. Adicionalmente, provee de clasificadores de Redes Bayesianas, también admite conjuntos de datos discretos y continuos (Scutari, 2010).

4.2.2 Algoritmos de aprendizaje de estructuras

El aprendizaje de estructuras consiste en encontrar las relaciones de dependencia entre las variables, de forma que se pueda determinar la topología o estructura de la red Bayesiana (Sucar, 2006).

Existen dos tipos generales de algoritmos de aprendizaje de estructuras: los basados en restricciones, los basados en puntajes y los híbridos. En el presente trabajo se usan algoritmos de aprendizaje basados en puntajes, pero de todas maneras se listarán todos los disponibles por la librería *bnlearn* de *R*.

Algoritmos basados en restricciones que buscan dependencias condicionales entre las variables para construir la estructura:

- Grow-Shrink (*GS*)
- Asociación Incremental Markov Blanket (*IAMB*)
- Asociación Incremental Rápida (*Fast – IAMB*)

- Asociación Incremental Interleaved (*Inter – IAMB*)
- Max-Min Parents & Children (*MMPC*)
- Hiton-PC semi-intercalada (*SI – HITON – PC*)

Algoritmos basados en puntajes estiman la estructura del modelo más probable considerando las posibles estructuras gráficas como distintos modelos, de tal manera que el problema se convierte en uno de maximizar algún score que califica los distintos modelos, es decir, se define un $Score(\mathcal{G}, \rho)$, donde \mathcal{G} , es una gráfica y ρ una distribución de probabilidad conjunta que se factoriza sobre \mathcal{G} , donde intenta resolver el problema o aproximar una solución: $\max_{\mathcal{G}, \rho} score(\mathcal{G}, \rho)$

- Escalada simple (*HC*)
- Tabu Search (*Tabu*)

Adicionalmente existen algoritmos híbridos que son una mezcla de métodos basados en restricciones y basados en puntajes:

- Max-Min Hill Climbing (*MMHC*)
- Maximización restringida general de 2 fases (*RSMAX2*)

4.2.2.1 Algoritmo Hill-Climbing (HC)

En el presente trabajo se utilizó el algoritmo de escalada simple (*HC*) que es una técnica que permite solo ir mejorando la solución repetidamente hasta que se maximiza una condición y comienza a partir de un punto en el espacio de búsqueda, y si el nuevo punto es mejor, se transforma en el punto actual, de caso contrario, selecciona un punto vecino y lo evalúa. El método termina cuando no hay mejorías o cuando se alcanza un número establecido de iteraciones.

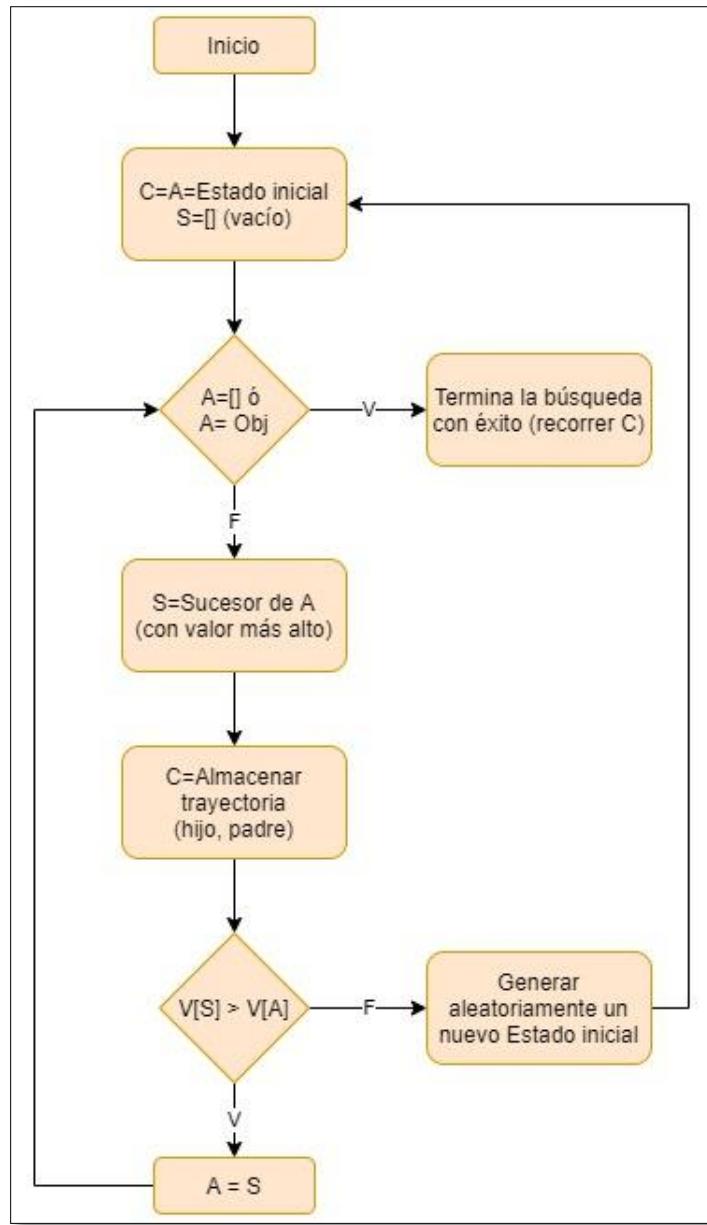


Figura 4: Algoritmo de Hill-Climbing.

4.2.2.2 Aprendizaje basado en medida de puntuación

El aprendizaje de estructuras se afronta con algoritmos basados en medidas de puntuación para resolver el problema de optimización. A continuación, se describen los utilizados en esta investigación: *Likelihood Score*, *Bayesian Information Criterion*, *Akaike Information Criterion*, *K2*, *Bdeu*, *TabuSearch*.

i. **Likelihood:**

Se basa en la función de verosimilitud que mide la probabilidad del conjunto de entrenamiento dado el modelo. Se define como la dupla compuesta por la estructura del modelo y el conjunto de parámetros dado el grafo, es decir $(\mathcal{G}, \theta_{\mathcal{G}})$ donde la máxima verosimilitud se expresa como:

$$\begin{aligned}\max_{\mathcal{G}, \theta_{\mathcal{G}}} L((\mathcal{G}, \theta_{\mathcal{G}}); \mathcal{D}) &= \max_{\mathcal{G}} \left[\max_{\theta_{\mathcal{G}}} L((\mathcal{G}, \theta_{\mathcal{G}}); \mathcal{D}) \right] \\ &= \max_{\mathcal{G}} [L((\mathcal{G}, \theta_{\mathcal{G}}^{MLE}); \mathcal{D})]\end{aligned}$$

Se concluye que, la estructura \mathcal{G} final, perteneciente al espacio de búsqueda, es aquella que maximiza la verosimilitud utilizando los parámetros obtenidos mediante *MLE*. La utilización de este método provoca un sobreajuste (overfit) de la estructura de entrenamiento, en la que se beneficia la red más compleja con respecto a la más sencilla (Ramírez Hereza & Ramos Castro, 2020)

ii. **Akaike Information Criterion:**

El criterio de información de Akaike (*Akaike Information Criterion o AIC*) selecciona el modelo que minimiza la probabilidad negativa penalizada por el número de parámetros como se especifica en la ecuación:

$$AIC = -2 \log p(L) + 2p$$

Donde L se refiere a la probabilidad bajo el modelo ajustado y p es el número de parámetros en el modelo. Este criterio tiene como objetivo encontrar el modelo que mejor se aproxime al proceso de generación de datos verdaderos desconocidos (Akaike, 1998)

iii. **Bayesian Information Criterion:**

El criterio de información bayesiano (*Bayesian Information Criterion o BIC*) se deriva dentro de un marco Bayesiano como una estimación del factor de Bayes para dos modelos competidores (Schwarz, 1978) (Kass & Raftery, 1995). *BIC* se define como:

$$BIC = -2 \log p(L) + p \log (n)$$

BIC difiere de *AIC* solo en el segundo término, que ahora depende del tamaño de la muestra n . *BIC* está diseñado para encontrar el modelo más probable dado los datos, pero penaliza las estructuras con mayor complejidad seleccionando el modelo con el *BIC* más bajo

- iv. **K2:** El algoritmo está basado en la optimización de una medida que es usada para explorar el espacio de búsqueda formado por todas las redes que contienen las variables de la base de datos. Se parte de una red inicial que se va modificando al añadir arcos, borrándolos o cambiando su dirección, obteniendo una red con mejor medida. En concreto, la medida K2 (Cooper & Herskovits, 1992) para una red G y una base de datos D es la siguiente:

$$f(G:D) = \log P(G) + \sum_{i=1}^n \left[\sum_{k=1}^{s_i} \left[\log \frac{\Gamma(\eta_{ik})}{\Gamma(N_{ik} + \eta_{ik})} + \sum_{j=1}^{r_i} \log \frac{\Gamma(N_{ijk} + \eta_{ijk})}{\Gamma(\eta_{ijk})} \right] \right]$$

Donde, N_{ijk} es la frecuencia de las configuraciones encontradas en la base de datos D de las variables x_i , donde n es el número de variables, tomando su j -ésimo valor y sus padres en G tomando su k -ésima configuración, donde s_i es el número de configuraciones posibles del conjunto de padres y r_i es el número de valores que puede tomar la variable x_i . Además, $N_{ik} = \sum_{j=1}^{r_i} N_{ijk}$ y Γ es la función Gamma.

A continuación se presenta el pseudo código del algoritmo K2:

```

procedure K2;
  {Input: A set of n nodes, an ordering on the nodes, an upper bound u on the number of parents a node may have, and a database D containing m cases}
  {Output: For each node, a printout of the parents of the node}
  for i := 1 to n do
     $\pi_i := 0;$ 
     $P_{old} := g(i, \pi_i);$  {This function is computed using equation (12).}
    OKToProceed := true

    while OKToProceed and  $|\pi_i| < u$  do
      let z be the node in  $Pred(X_i) - \pi_i$  that maximizes  $g(i, \pi_i \cup \{z\});$ 
       $P_{new} = g(i, \pi_i \cup \{z\});$ 

      if  $P_{new} > P_{old}$  then
         $P_{old} := P_{new};$ 
         $\pi_i = \pi_i \cup \{z\}$ 
      else
        OKToProceed := false;
    end {while};

    write('Node:',  $X_i$ , 'Parents of this node:',  $\pi_i$ )
  end {for};
end {K2};

```

- v. **BDeu (Bayesian Dirichlet Equivalent):** Asume uniformidad sobre las probabilidades de cada X_i y sus padres, es decir, sobre la distribución a priori debido a la falta de información previa. Produce el mismo valor para DAGs que contienen una distribución de probabilidad equivalente (Heckerman, Geiger, & Chickering, 1995)

$$BDeu(B, T) = \log(P(B)) + \sum_{i=1}^n \sum_{j=1}^{q_i} \left(\log \left(\frac{\Gamma(\frac{N'}{q_i})}{\Gamma(N_{ij} + \frac{N'}{q_i})} \right) + \sum_{k=1}^{r_i} \log \left(\frac{\Gamma(N_{ijk} + \frac{N'}{r_i q_i})}{\Gamma(\frac{N'}{r_i q_i})} \right) \right)$$

Que aparece cuando:

$$P(X_i = x_{ik}, \prod_{X_i} = \omega_{ij} | G) = \frac{1}{r_i q_i}$$

Esta puntuación solo depende de un parámetro, el tamaño de muestra equivalente N' .

Dado que no existe una regla generalmente aceptada para determinar los hiper parámetros $N'_{x_1 \dots x_n}$, no hay un buen candidato en particular para N' .

En la práctica, la puntuación BDeu es muy sensible con respecto al tamaño de muestra equivalente N' , por lo que se intentan varios valores.

vi. **TabuSearch (Búsqueda Tabú):** Es un método de optimización matemática de búsqueda por entornos caracterizada por dos aspectos:

- a. Enfrentar el problema de ciclos impidiendo temporalmente movimientos que podrían volver a la solución que ha sido recientemente visitada, es decir, generación de entornos restringidos.
- b. Emplea mecanismos de reinicialización para mejorar la capacidad del algoritmo para la exploración del espacio de búsqueda, es decir intensificación y diversificación.

Para llevar a cabo estos dos aspectos debe hacer uso de dos estructuras de memoria adaptativas: memoria a corto plazo o lista Tabú y memoria a largo plazo.

La memoria a corto plazo guarda información que guía la búsqueda desde el comienzo del procedimiento generando entornos restringidos. Para la memoria a largo plazo esta guarda la información que guiará la búsqueda a posteriori después de una primera etapa en la que se ha realizado una o varias ejecuciones del algoritmo. La información guardada en esta memoria es usada para comenzar con la búsqueda de una nueva solución de acuerdo con intensificar la búsqueda volviendo a visitar zonas prometedoras o diversifica la búsqueda visitando nuevas zonas no exploradas.

El algoritmo de búsqueda Tabú simple:

```

Generar solución inicial  $x_0$ 
 $k := 1$ 
 $x = x_0$  (x es la solución actual)
MIENTRAS la condición de finalización no se encuentre
HACER:
    Identificar  $N(x)$  (Vecindario de x)
    Identificar  $T(x, k)$  (Lista Tabú)
    Identificar  $A(s, k)$  (Conjunto de Aspirantes)
    Determinar  $N * (x, k) = \{(N(x) - T(x, k)\} \cup A(x, k)$  (Vecindario reducido)
    Escoger la mejor  $x \in N * (x, k)$ 
    "Guardar"  $x$  si mejora la mejor solución conocida  $x_k := x$ .
    Actualizar la lista tabú
     $k := k + 1$ 
FIN MIENTRAS

```

4.2.3 Algoritmos de aprendizaje de parámetros

Los algoritmos de aprendizaje de parámetros generan las estimaciones de parámetros para las tablas de probabilidades condicionales en cada nodo. Estos algoritmos requieren de una estructura aprendida que represente a un gráfico acíclico dirigido de caso contrario no es posible estimar puesto que deben estar especificados sus arcos. El método *bn.fit()* de la librería *bnlearn* de *R* se ajusta a los parámetros de una red Bayesiana dada su estructura y un conjunto de datos y devuelve la estructura subyacente a una red Bayesiana ajustada. Este método utiliza la *estimación de parámetros de máxima verosimilitud (mle)* o la *estimación de parámetros bayesianos (bayes)* con la salvedad que esta última solo está implementada para datos discretos y *mle* produce estimaciones para nodos gaussianos discretos y condicionales.

En una red Bayesiana se tiene dos casos en el aprendizaje:

- *Nodo raíz:* Se estima la probabilidad marginal. Por ejemplo: $P(A_i) \sim \frac{NA_i}{N}$, donde NA_i es el número de ocurrencias del valor i de la variable A , y N es el número total de casos o registros.
- *Nodos hoja:* Se estima la probabilidad condicional de la variable dados sus padres. Por ejemplo: $P(B_i | A_j, C_k) \sim \frac{NB_i A_j C_k}{NA_j C_k}$, donde $NB_i A_j C_k$ es el número de casos en que

$B = B_i$, $A = A_j$ y $C = C_k$ y NA_jC_k es el número de casos en que $A = A_j$ y $C = C_k$.
 (Sucar, 2006)

En la práctica, los datos a veces no están completos porque hay valores faltantes, es decir, faltan valores en una de las variables, o existen nodos ocultos donde faltan todos los valores de una variable, pero en el primer caso hay alternativas de solución como, por ejemplo: eliminar los registros con valores faltantes, tomar el valor promedio y reemplazar los faltantes por este promedio, entre otros. También, otro problema común es que existan nodos que no están conectados, por lo que, a través del conocimiento experto se deberían conectar antes de aplicar el aprendizaje de parámetros.

4.2.3.1 Estimador de Máxima Verosimilitud (MLE)

El estimador de máxima verosimilitud (*Maximum Likelihood Estimator* o *MLE*) maximiza la probabilidad de los parámetros de las funciones de densidad que dependen de la distribución de probabilidad y las observaciones de la muestra. Es decir, se busca que el modelo sea consistente con los datos y para ello se siguen los siguientes pasos.

- i. *MLE* define una función de máxima verosimilitud que es representada matemáticamente como:

$$L(\theta|x) = \prod_{i=1}^n f(x_i, \theta)$$

Donde x corresponde a la muestra $x = (x_1, \dots, x_n)$ y θ a los parámetros $\theta = (\theta_1, \dots, \theta_n)$, cuando más grande es el valor de $L(\theta|x)$ más probables serán los parámetros basados en la muestra.

- ii. Se calcula el logaritmo para encontrar las estimaciones de *MLE* derivando los productos de funciones de densidad:

$$\log L(\theta|x) = \sum_{i=1}^n \log f(x_i, \theta)$$

Las propiedades de los logaritmos permiten expresar la multiplicación del paso anterior como la sumatoria de logaritmos aplicados a la función de densidad.

iii. Maximizar la función logarítmica:

$$\theta^{MLE} = \arg \max_{\theta} L(\theta, x_1, \dots, x_M)$$

4.2.4 Algoritmos de inferencia

Una vez que se tiene la estructura y las estimaciones a priori de la red se pueden realizar las inferencias (probabilidad a posteriori) y esta puede hacerse en cualquier dirección dentro de la red. Conceptualmente cuando se utilizan evidencias y observaciones para establecer que una suposición sea cierta, se le denomina Inferencia Bayesiana. Esta observa la evidencia y calcula un valor estimado según el grado de creencia planteado en una hipótesis, es decir, que al tener una gran cantidad de datos se podrá obtener mejores resultados.

La probabilidad tiene dos formas de interpretación, una frecuentista y otra bayesiana, la primera es la probabilidad como la frecuencia relativa de un experimento aleatorio, donde, si el experimento se realiza un número grande de veces la probabilidad estaría dada por $P(A) = \lim_{n \rightarrow \infty} \frac{n_A}{n}$, donde n_A es el número de observaciones dentro del experimento que tiene el atributo y y n la cantidad de veces que se repite el experimento.

Para esta investigación se hace uso del enfoque Bayesiano que interpreta la probabilidad de manera subjetiva y está estrechamente relacionada con la probabilidad condicional determinada por el Teorema de Bayes. Al aplicar Inferencia Bayesiana se pueden identificar patrones, investigar sobre los efectos y condiciones, también se puede estudiar y analizar las deficiencias futuras de la información en estudio (Mesa Páez, Rivera Lozano, & Romero Davila, 2011).

Para hacer uso del enfoque Bayesiano la librería *bnlearn* de *R* proporciona un método llamado *cpquery* que permite consultar la probabilidad condicional de $P(A|E)$ un evento A dada una evidencia E , donde A y E corresponden a dos expresiones que describen el evento de interés y la evidencia condicionante en un formato tal que, el conjunto de datos del que se aprendió la red devuelva las observaciones correctas. Las estimaciones

devueltas son aproximaciones y no sumaran 1. Por defecto, el número de muestras aleatorias que se generan al mismo tiempo es de 10^4 . El método usado en el estudio es la ponderación de verosimilitud (*lw*) que es un algoritmo de inferencia aproximado basado en el muestreo de Monte Carlo (Scutari, 2010).

En el caso de *Python* el método de inferencia utilizado es mediante la eliminación de variables que corresponde a un algoritmo que desecha variables que no tiene ninguna relación de dependencia con su antecesor en la red. El algoritmo tiene una complejidad de tiempo exponencial en el número de variables, esto se puede reducir utilizando métodos heurísticos de ordenamiento para determinar la siguiente variable a considerar y así intentar minimizar el costo de la operación. Un método heurístico de ordenamiento habitual es moverse desde las hojas hacia arriba en la estructura Bayesiana.

4.2.4.1 Algoritmo de Eliminación de Variables

El algoritmo cuenta con una serie de pasos que consisten en determinar las variables irrelevantes que serán eliminadas mediante el orden de estas, agrupando las tablas dado un conjunto de factores, multiplicándolas para normalizar los valores finales del proceso de forma que la suma de dichos valores pase a ser 1.

```
FUNCION INFERENCIA_ELIMINACION_VARIABLES (X, e, RED)
    i. Sea RED_E el resultado de eliminar de RED las variables irrelevantes
       para la consulta realizada
    ii. Sea FACTORES igual a vacío
    iii. Sea VARIABLES el conjunto de variables de RED_E
    iv. Sea VAR_ORD el conjunto de VARIABLES ordenado según un orden de
        eliminación
    v. PARA cada VAR en VAR_ORD HACER
        a. Sea FACTOR el factor correspondiente a VAR (respecto de e)
        b. Añadir FACTOR a FACTORES
        c. Si VAR es una variable oculta hacer FACTORES igual a AGRUPA
           (VAR, FACTORES)
    vi. Devolver NORMALIZA(MULTIPLICA(FACTORES))
```

4.2.5 Funciones de validación de modelos (métricas)

Tras haber aprendido e inferido una red Bayesiana se puede tener la oportunidad de validar el cómo se ajusta el modelo a un conjunto de datos. La manera común de realizar la validación de modelos es a través de “validación cruzada” (*cross-*

validation), donde se estima el modelo con muestras aleatorias entre un 70% a 80% para entrenar los modelos y un 30% o 20% para testear el modelo. En la medida en que los modelos se ajustan a los datos se puede decir que existe evidencia de la validez de estos.

Para llevar a cabo una validación de modelos se recurren a ciertas mediciones que permiten buscar que estos tengan una cercanía a la realidad y en el caso de este estudio el foco está en estudiar las siguientes: Accuracy, Balanced Accuracy, Precision, Recall, ROC – AUC. Pero antes de entrar en detalle se debe tener conocimiento sobre la llamada Matriz de Confusión.

4.2.5.1 Matriz de Confusión

Tanto en inteligencia artificial como el aprendizaje automático la matriz de confusión es una herramienta que permite ver el desempeño de los algoritmos de aprendizaje, dónde cada columna representa el número de predicciones de cada clase y cada fila las instancias en la clase real, es decir, nos muestra qué tipo de aciertos y errores tienen los modelos cuando son sometidos a un aprendizaje automático.

VALORES PREDICCIÓN	Verdaderos Positivos VP	Falsos Positivos FP
	Falsos Negativos FN	Verdaderos Negativos VN
	VALORES REALES	

Tabla 1: Matriz de Confusión Binaria.

4.2.5.2 Accuracy Score (Puntuación de Exactitud)

La función de exactitud (Accuracy) es una métrica para evaluar modelos de clasificación, donde, la exactitud es la fracción de predicciones que el modelo realizó correctamente. En estadística, la exactitud está relacionada con el sesgo de una estimación y se representa de la siguiente manera:

$$Accuracy = \frac{VP + VN}{VP + VN + FP + FN}$$

Dónde:

- VP = Verdaderos Positivos
- VN = Verdaderos Negativos
- FP = Falsos Positivos
- FN = Falsos Negativos

Esta métrica, por si sola, no muestra un panorama completo cuando se está frente a un *conjunto de datos desequilibrados*, donde hay una gran disparidad entre las etiquetas positivas y las negativas.

4.2.5.3 Balanced Accuracy Score (Puntación de Exactitud Equilibrada)

Balanced Accuracy es una métrica usada cuando existen datos desequilibrados, representando las clases de resultados tanto positivos como negativos y no induce a error con datos desequilibrados como lo es cuando se usa la función de exactitud (Accuracy)

$$Balanced\ Accuracy = \frac{\frac{VP}{VP+FN} + \frac{VN}{VN+FP}}{2}$$

Dónde:

$\frac{VP}{VP+FN}$ Corresponde a la fórmula de *Sensibilidad o Tasa Positiva Verdadera* y $\frac{VN}{VN+FP}$

Corresponde a la fórmula de *Especificidad o Tasa Negativa Verdadera*.

En términos simples, Balanced Accuracy es solo el promedio de sensibilidad y especificidad

4.2.5.4 Precision Score

Esta métrica se refiere a cuan cerca se está del resultado de una predicción de valor verdadero. También, se refiere a la dispersión del conjunto de valores a partir de

mediciones repetidas de una magnitud, es decir, entre menor sea la dispersión mayor es la precisión.

$$Precision = \frac{VP}{VP + FP}$$

Dónde:

- VP = Verdaderos Positivos
- FP = Falsos Positivos

4.2.5.5 Recall Score (Puntuación de Recuperación)

La exhaustividad o sensibilidad se conoce como *la tasa de Verdaderos Positivos (VP)* que corresponde a la proporción de casos positivos que fueron correctamente identificados.

$$Recall = \frac{VP}{VP + FN}$$

Dónde:

- VP = Verdaderos Positivos
- FN = Falsos Negativos

4.2.5.6 Curva ROC - AUC

La curva de características operativa del receptor es una gráfica que muestra el rendimiento de un modelo y se representa por dos parámetros: Tasa de verdaderos Positivos y Tasa de Falsos Positivos. La tasa de verdaderos positivos (TPR) es sinónimo de exhaustividad y se define de la siguiente manera:

$$TPR = \frac{VP}{FP + VN}$$

Y la tasa de falsos positivos (FPR) se define de la siguiente manera:

$$FPR = \frac{FP}{FP + VN}$$

Entonces la curva ROC es la representación de TPR frente a FPR, dónde, reducir el umbral de clasificación clasifica más elementos como positivos por lo que aumentarán tanto los falsos positivos como los verdaderos positivos.

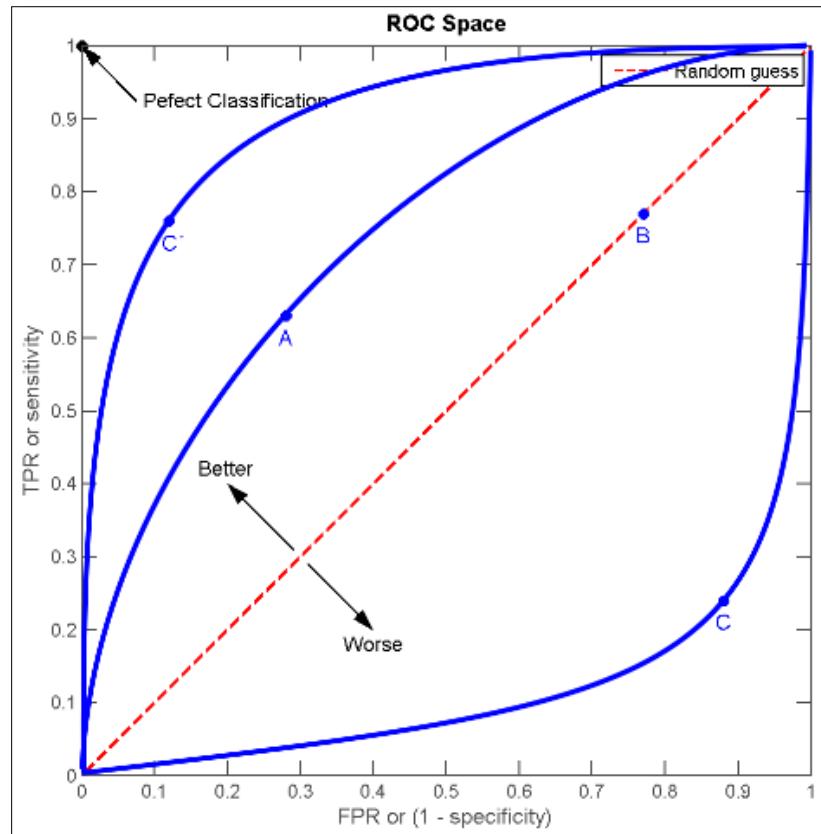


Figura 5: Curva ROC.

En la figura 5 se puede ver que las clasificaciones sobre la línea diagonal son mejores (TPR), es decir, mientras más cercanos a 1 mejor, pero bajo ella (FPR) tiene a representar una mala clasificación, siendo la línea diagonal una línea de estimación del 50%.

El área bajo la curva ROC (AUC) mide toda el área bidimensional por debajo de la curva ROC completa proporcionando una medición del rendimiento en todos los umbrales de clasificación posibles. Se puede interpretar como la probabilidad del modelo para clasificar un valor positivo aleatorio más alto que un valor negativo aleatorio. AUC oscila entre los valores 0 y 1, donde un modelo con predicciones son un 100% incorrectas tiene un AUC de 0 y si son 100% correctas su AUC es de 1.

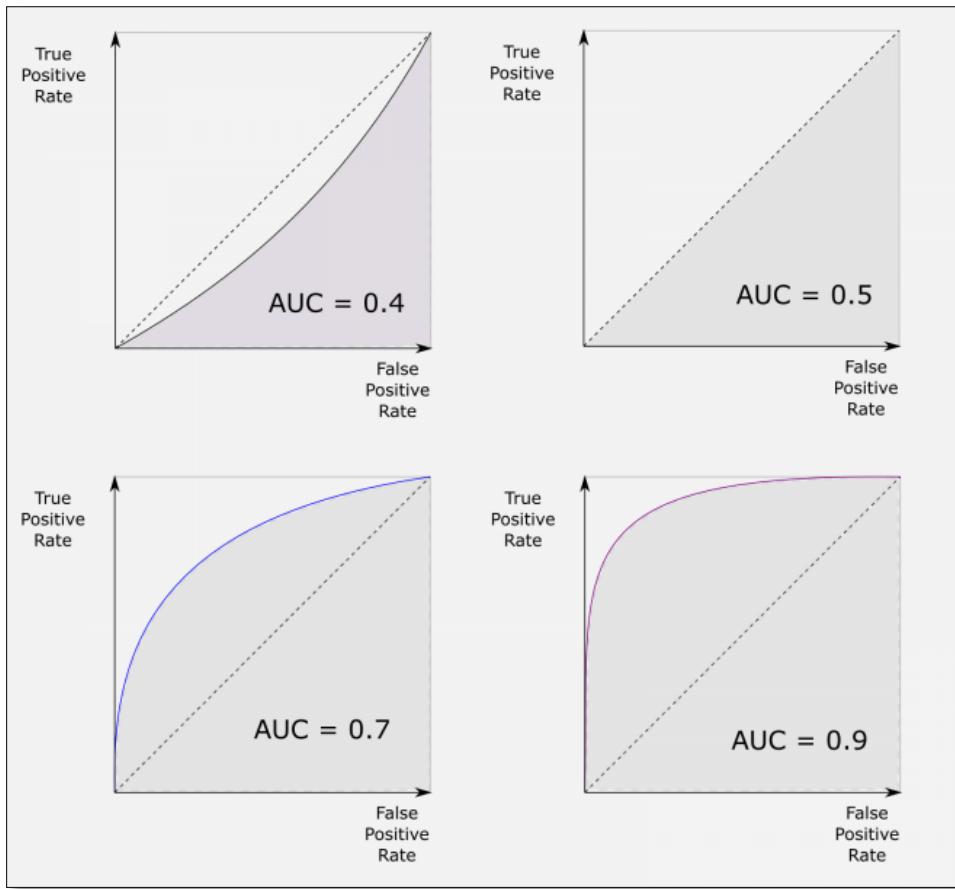


Figura 6: Área bajo la curva (AUC).

4.2.6 Método de Selección de Variables

Los métodos de selección de variables se utilizan para la reducción de dimensionalidad en conjuntos de muestras, ya sea para mejorar la precisión o aumentar el rendimiento de un modelo. La selección de variables es un proceso en el que automáticamente selecciona aquellas características que más contribuyen a la variable de predicción o salida en la que se está interesado. Al tener características irrelevantes en los datos puede disminuir la precisión de muchos modelos, sobre todo los algoritmos lineales.

La utilización de los métodos de selección de variables nos proporciona tres beneficios principales: *Reduce el sobreajuste* lo que implica tener menos datos redundantes para no tomar decisiones basadas en ruido, *Mejora la precisión* lo que implica menos datos

engañosos que afectan la precisión y *Reduce el tiempo de entrenamiento* lo que implica que los algoritmos entran más rápido.

Existen variados métodos de selección de características de los que se pueden nombrar, eliminación de variables por medio de baja varianza, por medio de puntuaciones altas, por medio de porcentaje de puntuaciones altas. En este apartado el foco estará en el método de selección de variables por medio de puntuaciones altas o más conocido como SelectKBest de la librería sklearn. Este método de selección de variables hace uso de una medida de puntuación que permite devolver puntuaciones univariantes y valores-p (Buitinck, y otros, 2013). Las funciones de puntuación disponibles para ser usados en este método, tenemos: Valor F de ANOVA (f_classif), Chi Cuadrado (chi2), Información Mutua (mutual_info_classif) que son métodos de clasificación y se describen a continuación:

- f_classif: Análisis de varianza es un método de puntuación, apropiado para entradas numéricas y datos categóricos, llamado en ocasiones análisis de varianza de Fisher, que surge de los conceptos de regresión lineal que es un modelo matemático que se utiliza para aproximar la relación de dependencia entre una variable dependiente Y, las variables independientes Xi y un término aleatorio.
- chi2: Método que mide la dependencia entre variables estocásticas eliminando las características que tienen más probabilidades de ser independientes de la clase e irrelevantes para la clasificación. Chi-cuadrado analiza variables nominales o cualitativas, es decir, determina la existencia, o no, de dos variables independientes y que sean independientes significa que no tienen relación.
- mutual_info_classif: Método de información mutua (MI en inglés) es una medida de dependencia mutua entre las variables. Es igual a cero si y solo si dos variables aleatorias son independientes y los valores más altos significan una mayor dependencia. Se basa en métodos no paramétricos basados en la estimación de la medida de incertidumbre a partir de distancias de los vecinos más cercanos

4.2.7 Técnica de Sobremuestreo (SMOTE)

SMOTE (Técnica de sobremuestreo de minorías sintéticas) es una técnica de sobremuestreo de datos son utilizados a menudo cuando los conjuntos de datos se encuentran desequilibrados, es decir, si las categorías de clasificación no están igualmente representadas. Generalmente para evaluar las técnicas de sobremuestreo o submuestreo se usan métricas de validación de modelos, en este caso la que más representa la validez del uso de esta técnica es el uso del área bajo la curva (AUC) y la Característica Operativa del Receptor (ROC).

La técnica funciona sobremuestreando cada muestra de la clase minoritaria e introduciendo ejemplos sintéticos a lo largo de los segmentos de línea que unen cualquiera o todos los vecinos más cercanos de la clase k minoritaria. Dependiendo de la cantidad de sobremuestreo, los vecinos de los k vecinos mas cercanos se eligen al azar. (Chawla, Bowyer, Hall, & Kegelmeyer, 2002)

4.2.7.1 Algoritmo de SMOTE

Algorithm SMOTE(T , N , k)

Input: Number of minority class samples T ; Amount of SMOTE $N\%$; Number of nearest neighbors k

Output: $(N/100)*T$ synthetic minority class samples

1. (* If N is less than 100%, randomize the minority class samples as only a random percent of them will be SMOTEd. *)
2. **if** $N < 100$
 3. **then** Randomize the T minority class samples
 4. $T = (N/100) * T$
 5. $N = 100$
6. **endif**
7. $N = (\text{int})(N/100)$ (* The amount of SMOTE is assumed to be in integral multiples of 100. *)
8. k = Number of nearest neighbors
9. numattrs = Number of attributes
10. $\text{Sample}[]$: array for original minority class samples
11. newindex : keeps a count of number of synthetic samples generated, initialized to 0
12. $\text{Synthetic}[]$: array for synthetic samples
(* Compute k nearest neighbors for each minority class sample only. *)
13. **for** $i \leftarrow 1$ **to** T
 14. Compute k nearest neighbors for i , and save the indices in the nnarray
 15. Populate($N, i, \text{nnarray}$)
16. **endfor**
 $\text{Populate}(N, i, \text{nnarray})$ (* Function to generate the synthetic samples. *)
17. **while** $N \neq 0$
 18. Choose a random number between 1 and k , call it nn . This step chooses one of the k nearest neighbors of i .
 19. **for** $attr \leftarrow 1$ **to** numattrs
 20. Compute: $dif = \text{Sample}[\text{nnarray}[nn]][attr] - \text{Sample}[i][attr]$
 21. Compute: $gap = \text{random number between } 0 \text{ and } 1$
 22. $\text{Synthetic}[\text{newindex}][attr] = \text{Sample}[i][attr] + gap * dif$
 23. **endfor**
 24. $\text{newindex}++$
 25. $N = N - 1$
26. **endwhile**
27. **return** (* End of Populate. *)

End of Pseudo-Code.

(Chawla, Bowyer, Hall, & Kegelmeyer, 2002)

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 y 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 en una institución mexicana. 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 a los estudiantes. 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 entrenamiento, uno para predecir los resultados y el segundo para calcular su exactitud.

El primer experimento, que considera todos los 20 atributos, tuvo una exactitud de las predicciones de 69.4581%. El segundo experimento utiliza los 15 mejores atributos, y da como resultado 71.4286%. Para el tercer experimento se evalúan varias cantidades de

mejores atributos, resultando la cantidad de 12 atributos con la mejor exactitud, 72.9064%. Entre estos atributos los más relevantes fueron la cantidad de cursos reprobados, el tipo de curso, si tiene beca, el promedio actual y el promedio en media superior.

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 dejó disponible 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 evaluó 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. De los resultados se destacan los de la cuarta red, en la cual se relacionan los conocimientos concatenados entre las evaluaciones, y el resultado final, que es la aprobación o no de la materia. Aunque esta red no incluye toda la información que pudiera relacionarse al objetivo final, se acerca mucho a la realidad, ya que la probabilidad de aprobar la materia da 49% y la real es 48%. Adicionalmente se observa que, si un alumno aprobó la materia, la probabilidad de conocer las preguntas conceptuales de la evaluación 2 y la evaluación 1 es respectivamente del 55% al 59%, y de resolver bien los problemas está entre el 44% y 68%.

Los resultados obtenidos en las dos primeras redes 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 a partir de factores socioeconómicos, estudios cursados por el estudiante y resultados académicos.

La base de conocimiento es de 773 estudiantes matriculados en el periodo 2012-2013 en la Facultad de Ciencias de la Ingeniería de la Universidad Técnica Estatal de Quevedo en Ecuador.

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 18 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. A continuación, y para investigar sobre las posibles relaciones entre las variables, se utilizaron los algoritmos PC, K2 y EM.

El algoritmo K2 generó una red donde las variables Carrera y Curso influyen fuertemente sobre la variable Aprobar, así como estos también influyen sobre la deserción. Las variables socioeconómicas se relacionan entre ellas, pero sin influencia en la aprobación o deserción estudiantil. En cuanto a los datos, hay un 11% de probabilidad que un estudiante deserte durante el transcurso de la carrera debido a la mala elección de esta, y el 30% de estos estudiantes desertan en el segundo año de la carrera.

Con el algoritmo PC se presenta un grupo similar de variables, pero con diferentes conjuntos de relaciones. El algoritmo EM que usa dos clústeres, obtiene un nodo clase del cual depende las variables, y se evidencia la relación más fuerte con poseer servicio de tv cable y servicio de internet.

En cuanto a las variables socioeconómicas, se puede determinar que el 24% de los estudiantes viven en un domicilio diferente al de la familia, el 86% de estas viviendas son de tipo villa, y el propietario de la vivienda en un 33% es el padre. El 77% de los estudiantes no tienen servicio de tv cable, el 40% pueden conectarse a internet en sus hogares y 12% cuenta con plan celular. El 5% cuenta con vehículo propio, pero sólo el 3% llega a la Universidad en vehículo propio. Además, el 20% de estos estudiantes trabajan, teniendo como consecuencia que el 30% de los estudiantes reprueban, de los cuales solo el 11% desertan.

Los tres algoritmos tuvieron en común el agrupamiento de las variables Carrera, Curso, Aprobar y Deserción, así como también, la gran influencia que tiene la variable Carrera hacia las demás variables del conjunto.

Los resultados evidencian que un factor de impacto para la deserción estudiantil es el año en el que está cursando y los diferentes factores socioeconómicos 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)

Dado que el abandono en cursos iniciales de programación en estudios universitarios generalmente es alto y los índices de superación bajos, y además que las actividades prácticas se consideran uno de los elementos que más inciden positivamente en la eficacia del aprendizaje, es que nace el interés en esta investigación.

Este trabajo busca encontrar una relación entre el perfil de los estudiantes y su grado de participación en la asignatura introductoria de programación de la Universitat Oberta de Catalunya. 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. En el estudio se utilizaron 1043 registros académicos de 3 semestres consecutivos (2017-2; 2018-1; 2018-2), los que contienen datos sociodemográficos, de experiencia académica y semestre en curso.

El objetivo es relacionar las entregas que hacen los estudiantes con el resultado obtenido en la actividad obligatoria mediante el uso de redes Bayesianas, por lo cual se construyeron dos modelos diferentes para establecer la relación entre las variables. El primer modelo pretende relacionar las variables sociodemográficas y la experiencia académica con el hecho de iniciar la secuencia de actividades propuesta. El segundo modelo pretende establecer la relación causal entre el hecho de hacer o no cada una de las actividades propuestas con las siguientes, y con la actividad obligatoria final, usando también las variables identificadas como relevantes en el primer modelo. En ambos casos se usó un algoritmo de ascensión de colinas con el criterio de información bayesiano

(BIC) para extraer la estructura subyacente de la red a partir del conjunto de datos. Para la creación de las redes bayesianas se usó el paquete de R “bnlearn”.

La primera red muestra que la única variable que incide sobre el hecho de realizar la primera actividad propuesta es el grado que está cursando el estudiante. El perfil sociodemográfico del estudiante parece no tener ninguna relación causal con el hecho de realizar la primera actividad.

El segundo modelo se construye incluyendo la variable destacada en el modelo anterior como único antecedente de la primera actividad, mostrando que las posibles relaciones causales más importantes se dan entre actividades consecutivas, decreciendo a lo largo del tiempo. Las primeras actividades son las más importantes a la hora de conseguir que el estudiante se aproveche del diseño del curso.

Luego se realiza una estimación con evidencias firmes mediante una simulación de Monte Carlo, la cual muestra que la probabilidad de realizar la primera actividad varía según el grado, es de 0.8546 para Informática, 0.7532 para Telecomunicaciones, 0.9346 como complemento de formación de máster y 0.6531 como asignatura libre. La probabilidad de superar la actividad obligatoria por grado según la realización de la primera actividad muestra que el grado ya no tiene relación, en cambio la primera actividad si influye, por ejemplo, para el grado de Informática esta probabilidad es de 0.6372 dado que realiza la primera actividad. Los resultados también 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 de España, 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 de interés. Éstas consisten en datos personales, socioeconómicos y académicos, y fueron adaptadas para ser usados con el software Elvira.

El programa Elvira, llamado Entorno de Desarrollo para Modelos Gráficos Probabilísticos, consta de 3 modos básicos: Edición, Inferencia y Aprendizaje. Con este último se construyen las tablas de probabilidad y la estructura de la red Bayesiana haciendo uso del algoritmo PC de Spirtes, Glymour, and Scheines.

La red con todas las variables resulta compleja, por lo tanto, se elabora una nueva red eliminando variables y considerando sólo las que definen al alumno, quedando 16 variables en esta red.

A continuación, en el modo Inferencia de Elvira, se hace un análisis de variables mediante propagación de probabilidades para la obtención de los perfiles buscados, es decir, se calculan las probabilidades a posteriori de las variables de interés una vez observado el valor de otras variables.

En cuanto a los estudios de los padres, se calculan las probabilidades de los estudios de la madre fijando el estudio del padre, mostrando que en general el estudio de la madre es el mismo que el del padre. La probabilidad de que el estudio de la madre sea primario dado que el del padre es primario es de 0.7784169, la más alta.

Para estudiar la titulación escogida por el alumno se fijan variables socioeconómicas. En el caso del sexo del alumno, se concluye que este influye en la titulación elegida. Por ejemplo, la probabilidad de Psicología en los hombres es de 0.03952249 y en las mujeres de 0.10693198. Fijando el domicilio familiar, las probabilidades apenas varían al cambiar de zona. Por ejemplo, en el Distrito 8 Derecho tiene una probabilidad de 0.1059 y en el distrito Poniente de 0.1054.

Posteriormente, se realiza una inferencia abductiva, que es la búsqueda del perfil más probable de los individuos de una población, bajo determinadas condiciones impuestas por las variables observadas. Se obtienen los cuatro perfiles más probables de un alumno, compuestos por 7 variables, siendo las probabilidades a posteriori de

0.00146583, 0.00143001, 0.00142169 y 0.0013870, lo que indica que el alumnado forma un grupo muy heterogéneo.

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 (Minerva).

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, & Smyth, 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, & Smyth, 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.

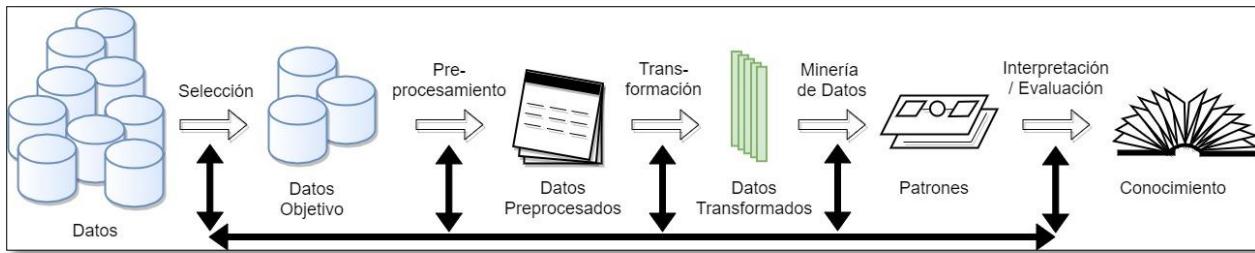


Figura 7: Proceso KDD (Fayyad, Piatetsky-Shapiro, & Smyth, 1996).

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).

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)

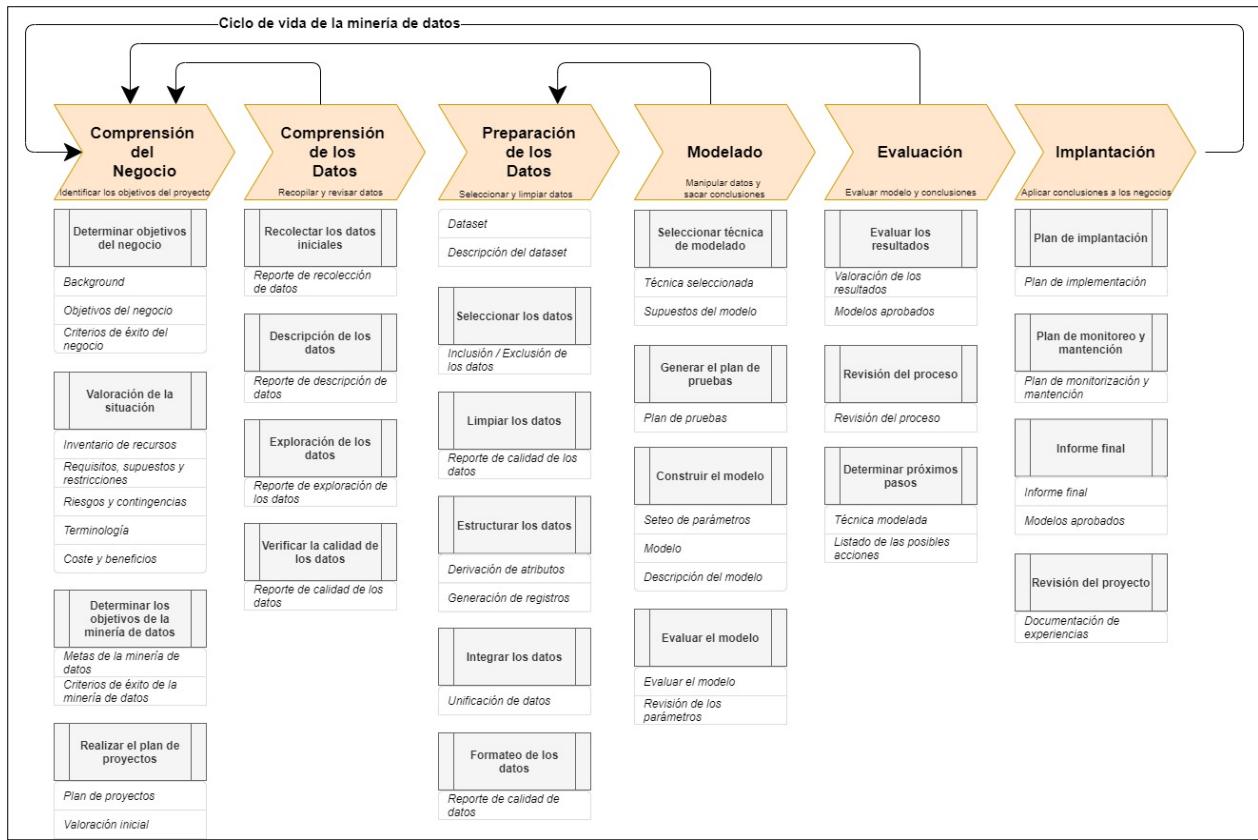


Figura 8: Metodología CRISP-DM.

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 (SAS 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).

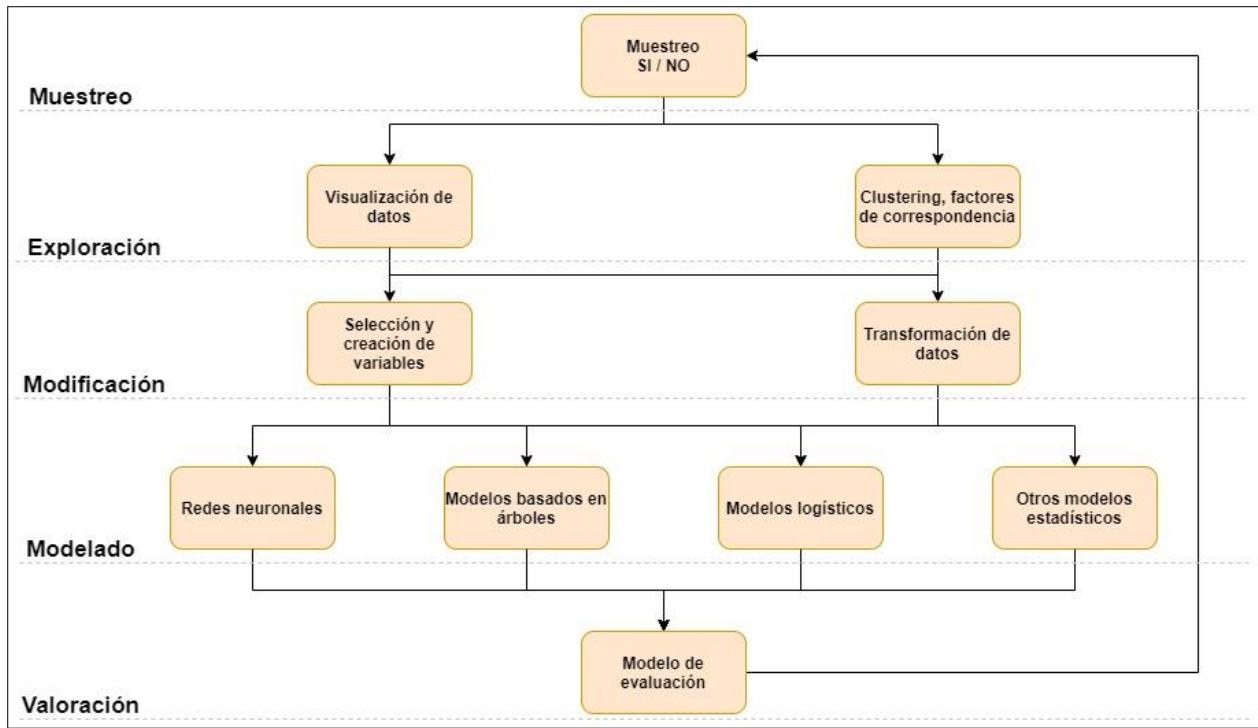


Figura 9: Metodología SEMMA.

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 que existen, para la búsqueda de patrones y la extracción de información que se ocultan en grandes cantidades de información, son:

- **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, posee 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 Management 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:

Inicio	Planificación	Ejecución	Monitoreo y Control	Cierre
	Gestión de los Interesados			
	Gestión de las Adquisiciones			
	Gestión de los Riesgos			
	Gestión de las Comunicaciones			
	Gestión de los Recursos			
	Gestión de la Calidad			
Gestión de los Costos		Gestión de los Costos		
Gestión del Tiempo		Gestión del Tiempo		
Gestión del Alcance		Gestión del Alcance		
	Gestión de la Integración			

Tabla 2: 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.

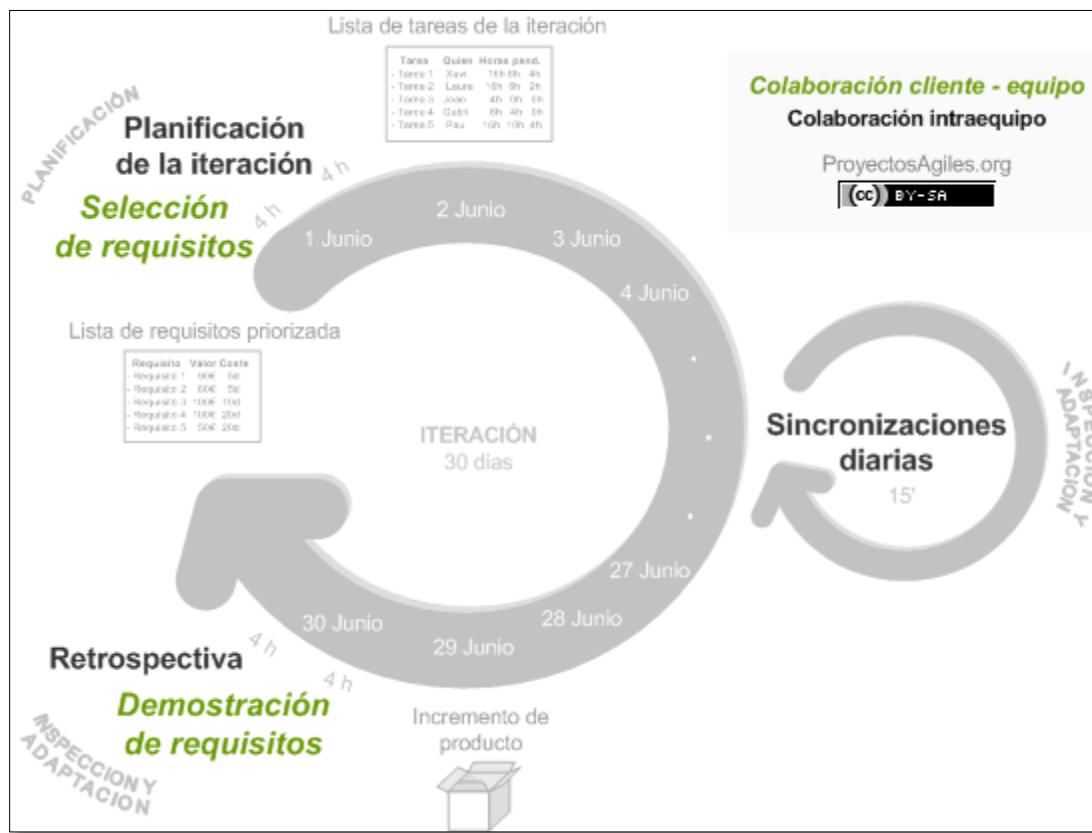


Figura 10: Proceso de Scrum.

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 3: 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.

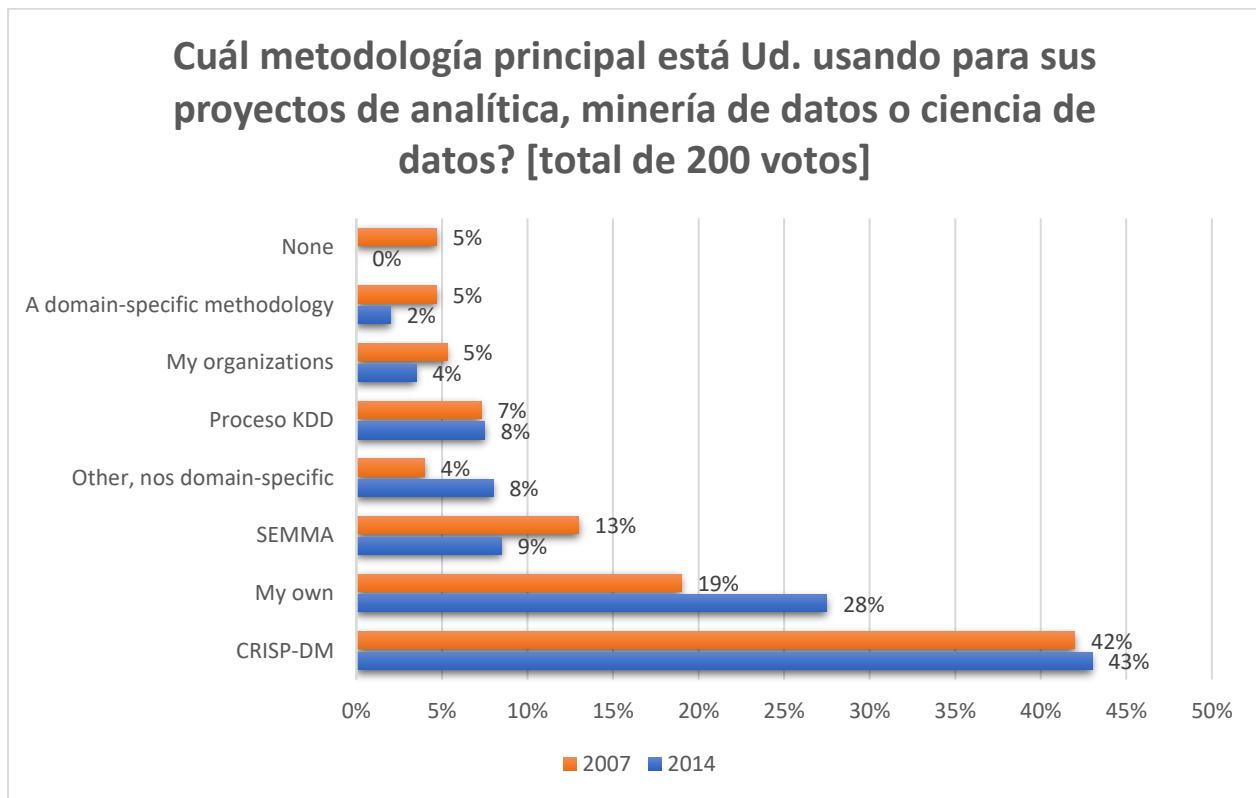


Figura 11: Encuesta KDnuggets sobre métodos análisis, minería de datos o ciencia de datos.

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, & Smyth, 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 dé 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 4: 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. En la siguiente tabla se presentan las tareas resumidas y sus fechas.

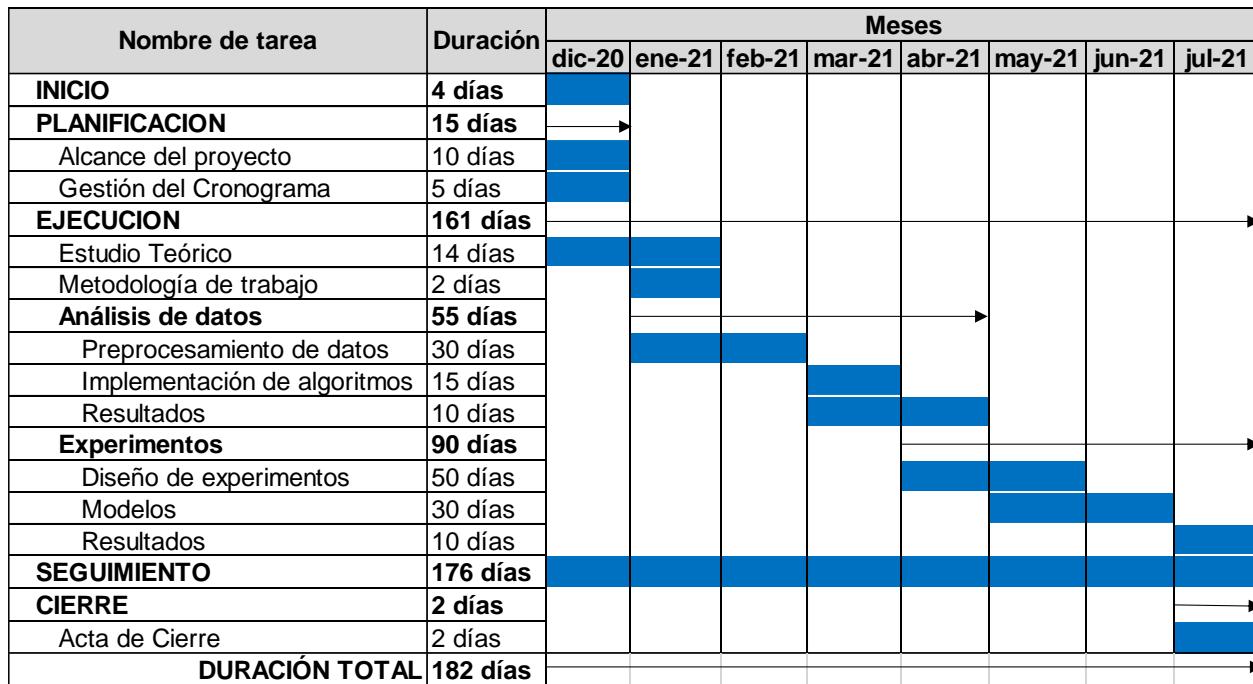


Tabla 5: Diagrama de Gantt.

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

- Se comienza revisando el tamaño y dimensión de la muestra de datos, para luego verificar si existen valores nulos y los tipos de datos presentes. Después, se identifican y descartan los atributos que no aportan información o no pueden ser analizados numéricamente. Luego, se realiza un análisis descriptivo de los datos que se conservan. Posteriormente, se hace conversión numérica a los atributos categóricos alfabéticos, y se calculan sus tasas de frecuencia.

Paso 2: Reducción y proyección de datos

- Se realiza análisis bidimensional de los atributos cuantitativos y tablas de contingencia para los atributos categóricos. Se seleccionan las variables relevantes del conjunto de datos según el análisis previo, y se procede a discretizar las variables continuas y las variables discretas que lo requieren por presentar alta variedad de valores.

Paso 3: Elección de los algoritmos de minería de datos

- Se analizan los algoritmos de aprendizaje de estructura de redes Bayesianas, que están disponibles en la librería a utilizar, para luego elegir el más conveniente. Dado que el algoritmo seleccionado es basado en métricas, se analizan los tipos de métricas disponibles para elegir la que se va a usar. Puesto que la clase de interés se encuentra desbalanceada, se elige la técnica de balanceo para su aplicación. Se define un conjunto de restricciones a aplicar, basado en experiencia, para generar otro experimento. Adicionalmente, se aplica una técnica de selección de variables para un tercer experimento.

Paso 4: Minería de datos

- Diseño de modelos
- Aplicación de los modelos
- Resultados de la aplicación de los modelos

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 objetivo del análisis contiene datos de 467 estudiantes, clasificados en 80 atributos. Luego de una revisión, se identifican 5 atributos que no aportan información y pueden ser descartados: “rut”, “usuario”, “correo”, “nombre” y “profesor”. La tabla 5 muestra la descripción y tipo de datos de estos y otros atributos eliminados.

En cuanto al juego diagnóstico, este consiste en 6 niveles consecutivos de evaluación, donde cada uno es de mayor complejidad al anterior. Entre los atributos relacionados a este juego se encuentran algunos redundantes, ya que para cada nivel se cuenta con tiempo de inicio, tiempo de finalización y tiempo total, donde este último corresponde a la sustracción de los dos anteriores. Debido a esto, los atributos “i1”, “f1”, “i2”, “f2”, “i3”, “f3”, “i4”, “f4”, “i5”, “f5”, “i6” y “f6” son descartados (ver tabla 5) y se considera sólo el tiempo total en los niveles. También, para cada nivel existe el atributo que representa la solución generada por el estudiante, y dado que este valor consiste en una secuencia de caracteres y números, y además existe el atributo del largo de la solución, se decide eliminar los 6 atributos de las soluciones generadas porque se considera que el largo de la solución aporta suficiente información de tipo numérica y cuantitativa.

Respecto al curso de programación, existen atributos relacionados a las evaluaciones parciales, los cuales son descartados debido a que la investigación está enfocada en indagar sobre el hecho de la reprobación del curso. Por lo tanto, los atributos “sol1”,

“sol2”, “sol3”, “sol4”, “tarea1”, “tarea2”, “controles”, “np” y “examen” también son eliminados (ver tabla 5).

Variable	Descripción	Tipo 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 6: Lista de atributos descartados del análisis

En total se eliminan 32 atributos, quedando 48 considerados para el análisis, que de ahora en adelante llamaremos variables. Estas pueden ser agrupadas, para facilitar su entendimiento, según estén relacionadas al juego de diagnóstico previo al curso, la prueba de diagnóstico de aptitudes y el curso de programación. En la tabla 6 se muestran las variables relacionadas al juego de diagnóstico.

Variable	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	Puntaje total de pruebas acumuladas de los 6 niveles	int64
game_score	Suma de lt + tt + pt	float64

Tabla 7: Variables relacionadas al juego de diagnóstico.

Como puede verse, cada nivel del juego tiene asociadas varias variables que pudieran reflejar el desempeño del estudiante en ese nivel. Respecto a este conjunto es relevante la variable “game_score”, debido a que cuantifica el puntaje final del estudiante en el juego, a partir de los resultados obtenidos en cada nivel de evaluación. Además, existen 12 variables binarias “op1”, “op2”, “op3”, “op4”, “op5”, “op6”, “sv1”, “sv2”, “sv3”, “sv4”, “sv5” y “sv6”, las cuales tienen asignadas 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, aunque no es calculada a partir de las otras variables.

Variable	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 8: 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”, ya que representa el hecho de si el estudiante aprobó o reprobó el curso de programación, por lo tanto, es la variable de interés para el modelo y sobre la cual se desea realizar las inferencias de causalidad.

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

Tabla 9: 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 deben ser 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 10: Conversión de valores en variable “programa”.

La variable “estado”, que es de tipo binaria, y que tiene relación a la situación del curso, es representada por el valor “A” para el caso de aprobado y “R” para el caso reprobado, entonces estos 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 11: 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]: 1 #abriendo el conjunto de datos desde un archivo CSV y asignandolo los datos al DataFrame "df"
2 df = pd.read_csv('dataset_a.csv', sep=';', error_bad_lines=False)
```

Figura 12: 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]: 1 #identificando las columnas con valores nulos "NaN"
2 null_columns=df.columns[df.isnull().any()]
3 df=null_columns.sum()
4 print(df[df.isnull().any(axis=1)][null_columns].head())
```

Empty DataFrame
Columns: []
Index: []

Figura 13: 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	resetsl1	pruebas1	largoSol1	tiempo2	resetsl2	pruebas2	largoSol2	tiempo3	resetsl3	pruebas3	largoSol3	tiempo4	resetsl4	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	
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 14: 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	reset5	pruebas5	largosol5	tiempo6	reset6	pruebas6	largosol6	optimal	optima2	optima3	optima4	optima5
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	
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 15: 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	0.396287	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	17.10026	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 16: 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	solemnel	solemne2	solemne3	solemne4	tareal	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 17: Análisis 1D de las últimas 14 variables.

Adicionalmente, se calcula la frecuencia de los valores en la variable categórica “programa” y la variable binaria “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%).

In [10]:	1 df['programa'].value_counts() / 467 * 100
Out[10]:	5 40.256959 4 29.336188 3 23.340471 2 6.638116 1 0.428266 Name: programa, dtype: float64
In [11]:	1 df['estado'].value_counts() / 467 * 100
Out[11]:	0 89.507495 1 10.492505 Name: estado, dtype: float64

Figura 18: 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”

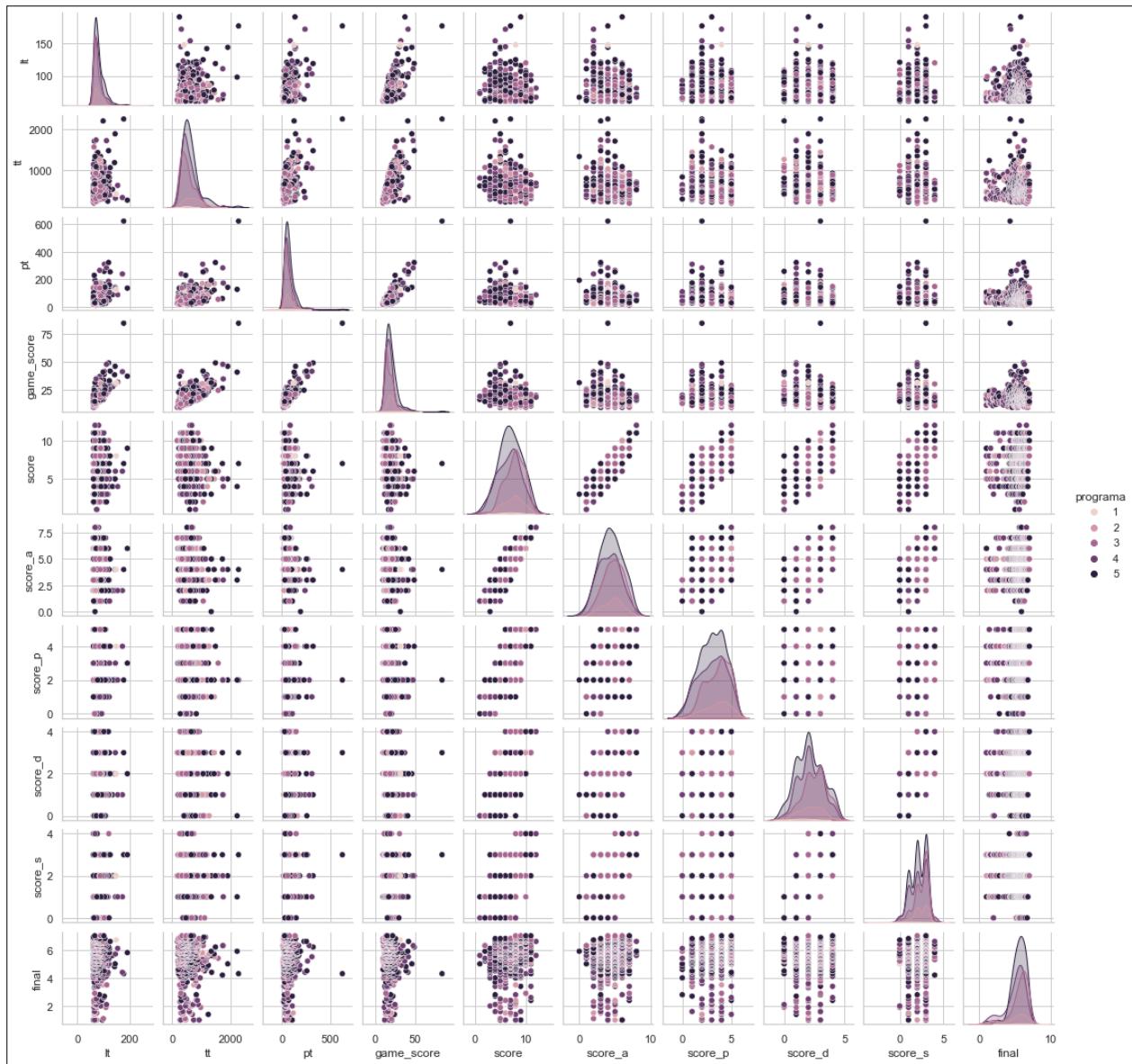


Figura 19: Matriz de correlación de variables.

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).

Programa	Estado			Total
	Aprobado	Reprobado		
Bachillerato en Ciencias	0,43	0,00		0,43
Ingeniería Industrial	5,14	1,50		6,64
Ingeniería Civil Informática	21,20	2,14		23,34
Ingeniería en Computación e Informática	24,41	4,93		29,34
Ingeniería Civil Industrial	38,33	1,93		40,26
Total	89,51	10,49		100,00

Tabla 12: 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.

op1	sv1		Total
	0	1	
0	2,14	42,18	44,32
1	0,00	55,67	55,67
Total	2,14	97,86	100,00

Tabla 13: Tabla de contingencia op1 / sv1.

op2	sv2		Total
	0	1	
0	5,78	24,41	30,19
1	0,00	69,81	69,81
Total	5,78	94,22	100,00

Tabla 14: Tabla de contingencia op2 / sv2.

op3	sv3		Total
	0	1	
0	60,17	0,00	60,17
1	0,00	39,83	39,83
Total	60,17	39,83	100,00

Tabla 15: Tabla de contingencia op3 / sv3.

		sv4		Total
		op4	0	1
op4	0	23,98	28,69	52,68
	1	0,00	47,32	47,32
Total		23,98	76,02	100,00

Tabla 16: Tabla de contingencia op4 / sv4.

		sv5		Total
		op5	0	1
op5	0	12,63	46,25	58,89
	1	0,00	41,11	41,11
Total		12,63	87,37	100,00

Tabla 17: Tabla de contingencia op5 / sv5.

		sv6		Total
		op6	0	1
op6	0	20,34	31,69	52,03
	1	0,00	47,97	47,97
Total		20,34	79,66	100,00

Tabla 18: Tabla de contingencia op6 / sv6.

6.3 Preprocesamiento de datos

Se realizó una selección de las variables relevantes en función de los análisis 1D y 2D previamente vistos, las cuales se muestran en la tabla 11. Dichas variables fueron las que mostraron una mayor correlación entre sí, indicando un mayor aporte de información con respecto a las demás.

Variable	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	Puntaje total de pruebas acumuladas de los 6 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	int64
final	Nota final del estudiante	float64
estado	Estado final del estudiante (Aprobado / Reprobado)	int32

Tabla 19: Listado de variables seleccionadas para realizar el modelo.

Luego se realizó la discretización de algunas variables (ver tabla 12), dado que o poseen datos continuos: “tt”, “game_score” y “final”, o siendo discretas poseen un rango de valores que no permitirían ejecutar de buena manera el modelo que se implementa en el siguiente paso: “lt” y “pt”.

Variables	Descripción	Tipo de datos
lt	Largo acumulado de las soluciones de los niveles	int64
tt	Tiempo total acumulado de los niveles	float64
pt	Puntaje total de pruebas acumuladas de los 6 niveles	int64
game_score	Suma de lt + tt + pt	float64
final	Nota final del estudiante	float64

Tabla 20: Lista de variables discretizadas.

Se utilizó 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, en las variables “lt”, “tt”, “pt” y “game_score” se usaron 5 “bins”, y en la variable “final” se usó una discretización de 7 “bins”. Como parámetro se usó la estrategia algorítmica llamada “k-means”, que corresponde a una clasificación no supervisada que agrupa objetos en k grupos basándose en sus características.

6.4 Experimentos

6.4.1 Diseño de experimentos

Para llevar a cabo el aprendizaje automático se construyó un programa en Python que permitió diseñar y llevar a cabo una serie de experimentos utilizando múltiples alternativas paramétricas para la utilización de distintos algoritmos que permiten realizar el aprendizaje de estructura para crear una Red Bayesiana completamente dirigida a partir de los datos utilizados en la investigación, y como también conocer las probabilidades a priori de la estructura aprendida por medio de algoritmos que permiten realizar el aprendizaje de estos. Con esos dos conocimientos, se pudo concretar la probabilidad conjunta de la clase en estudio, en este caso, conocer las probabilidades de aprobación / reprobación de un alumno en el curso de programación a partir de resultados de un juego y prueba de diagnóstico. Los siguientes son los pasos para llevar a cabo los experimentos:

Paso 1: Realizar una validación cruzada estratificada de 5 repeticiones en los datos.

Paso 2: Separar la información en una porción de entrenamiento y una de pruebas.

Paso 3: Aprender la estructura de los datos de entrenamiento y construir un Grafo Acíclico Dirigido (DAG), la Red Bayesiana.

Paso 4: Aprender los parámetros o probabilidades a priori de la Red Bayesiana.

Paso 5: Conocer la probabilidad conjunta por medio de inferencias para la porción de entrenamiento.

Paso 6: Obtener las métricas para la porción de entrenamiento.

Paso 7: Conocer la probabilidad conjunta por medio de inferencias para la porción de pruebas.

Paso 8: Obtener las métricas para la porción de pruebas.

Paso 9: Volver al paso 3 a la siguiente repetición de la validación cruzada hasta que se cumplan las 5 repeticiones.

En la búsqueda del modelo con mejor rendimiento se experimentó con la librería bnlearn implementada en Python y también con su implementación en R. Como se

descubrió en el análisis de datos inicial, la base de datos utilizada presentaba información sesgada ya que existe un 90% de registros con aprobaciones y un 10% con reprobaciones indicando un alto grado de desbalanceo de la información. Dado el escenario anterior, se experimentó con los datos desbalanceados y con la utilización de técnicas de balanceo de información.

Por consiguiente, se llevaron a cabo los siguientes experimentos que fueron siempre en la búsqueda de un mejor rendimiento:

- i. Experimentos bajo la librería bnlearn de Python con datos balanceados y desbalanceados.
 - a. Para estos experimentos se usaron las funciones de puntuación BIC, K2, BDEU
- ii. Experimentos bajo la librería bnlearn de R con datos balanceados y desbalanceados. En estos experimentos se notó que la librería bajo lenguaje R tiene un mayor potencial y flexibilidad a la hora de implementarla ampliando el número de experimentos usando los datos completamente *discretos* como *mixtos*, es decir una porción de variables discretas y el resto continuas.
 - a. Se ejecutaron experimentos con datos *discretos* usando las funciones de puntuación AIC, BIC, LOGLIK
 - b. Se realizaron experimentos con datos *mixtos* usando las funciones de puntuación AIC – CG, BIC – CG, LOGLIK – CG

De todos los experimentos realizados se optó, como candidatos a seguir experimentando, por aquellos que cumplían con un mejor rendimiento para llevarlos a un siguiente nivel utilizando técnicas de selección de variables, aplicables en múltiples de 5, partiendo de las 5 mejores hasta llegar a 20 variables. Además, la aplicación de restricciones semánticas al aprendizaje de estructuras por medio de blacklist o whitelist.

Adicionalmente, se diseñaron dos experimentos para realizar en la herramienta Weka, con el fin de comparar y confirmar los resultados de los experimentos mencionados anteriormente. Dichos experimentos son los siguientes:

- i. Generación de la red Bayesiana considerando sólo las variables del juego diagnóstico, para la búsqueda de relaciones entre ellas y la clase “estado”. Este modelo contiene 41 variables.
- ii. Generación de la red Bayesiana con las mismas variables del modelo en Python y R, incluyendo por lo tanto 24 variables.

Para discretizar las variables continuas se elige el método de “binning” simple con intervalos de igual frecuencia, tomando 5 bins. El método de intervalos de igual ancho no es útil para esta muestra de datos debido a la existencia de valores extremos.

Al igual que en los experimentos de Python y R, se realiza validación cruzada estratificada de 5 particiones y se balancean los modelos. Se prueban diferentes combinaciones de parámetros disponibles en Weka, incluyendo los algoritmos Hill Climber, Repeated Hill Climber, K2 y TabuSearch, en las modalidades global y local, con el objetivo de presentar el que funcione mejor. Como estimador de probabilidades se prueban el Estimador Simple con alfa 0.5 y el Estimador Simple con alfa 0.0 que equivale al Estimador de Máxima Verosimilitud (MLE). La técnica de balanceo a utilizar es Matriz de Costo.

La parametrización seleccionada para el algoritmo Hill Climber es la siguiente:

- i. Iniciar como Bayes Ingenuo: No
- ii. Aplicar corrección Markov Blanket: No
- iii. Máximo número de padres: 10
- iv. Usar inversión de arcos: Sí

Para la versión global se selecciona “K-fold” como tipo de validación cruzada, y para la versión local se eligen los tipos de métricas MDL (BIC) y AIC para las pruebas.

Respecto al algoritmo Repeated Hill Climber, los parámetros son:

- i. Número de corridas: 5 y 10
- ii. Semilla: 1

6.4.2 Modelo bajo Python

Dentro de la serie de experimentos realizados, los que son mencionados en este punto, fueron ejecutados con la utilización de librerías disponibles en el lenguaje de programación Python. Principalmente, la librería base utilizada (bnlearn), proporcionó un set de herramientas que permiten aprender de los datos e inferir una probabilidad conjunta. Adicionalmente, y dado el conocimiento adquirido sobre los datos, se utilizaron técnicas de balanceo de datos para ampliar la gama de experimentos y determinar las métricas que permitan determinar, bajo que escenarios, el modelo presenta una mejor validez. Los experimentos realizados fueron los siguientes:

- Con datos desbalanceados usando Criterio de Información Bayesiana (bic): Tiempo de ejecución aproximado: 9 horas 30 minutos.
- Con datos desbalanceados usando Equivalente Dirichlet bayesiano (bdeu): Tiempo de ejecución aproximado: 9 horas 30 minutos.
- Con datos desbalanceados usando Puntuación K2: Tiempo de ejecución aproximado: 11 horas 30 minutos.
- Con datos balanceados usando Criterio de Información Bayesiana (bic): Tiempo de ejecución aproximado: 3 horas.
- Con datos balanceados usando Equivalente Dirichlet bayesiano (bdeu): Tiempo de ejecución aproximado: 8 horas.
- Con datos balanceados usando Puntuación K2: Tiempo de ejecución aproximado: 4 horas.

Cabe destacar que el uso de la librería bnlearn de Python, los tiempos de ejecución en el cálculo de inferencias son extremadamente largos, recordando que la base de datos utilizada solo cuenta con 467 registros.

6.4.2.1 Resultados de experimentos de modelo bajo Python

La siguiente tabla muestra un resumen del promedio de las métricas obtenidas del **set de validación (test)** usando los datos **desbalanceados**:

Balanceado	Métricas	Medidas de Puntuación		
		BIC	K2	BDEU
NO	Accuracy	88,863%	88,863%	88,863%
NO	Balanced Accuracy	63,412%	63,412%	63,412%
NO	Precision Score	92,461%	92,461%	92,461%
NO	Recall Score	88,863%	88,863%	88,863%
NO	ROC AUC	59,695%	59,695%	59,695%
NO	Class Ratio	10,492%	10,492%	10,492%

Tabla 21: Resultados set de validación desbalanceado (modelos bajo Python).

Para los datos desbalanceados el uso de las medidas de puntuación no hace variar los resultados, esto puede deberse a que el algoritmo Hill – Climbing muestra un comportamiento similar en cada medida.

La siguiente tabla muestra un resumen del promedio de las métricas obtenidas del **set de validación (test)** usando las técnicas de **balanceo** de datos:

Balanceado	Métricas	Medidas de Puntuación		
		BIC	K2	BDEU
SI	Accuracy	84,802%	83,089%	85,230%
SI	Balanced Accuracy	59,418%	58,085%	56,186%
SI	Precision Score	10,492%	10,492%	10,492%
SI	Recall Score	83,627%	80,705%	84,550%
SI	ROC AUC	84,802%	83,089%	85,230%
SI	Class Ratio	66,498%	66,442%	69,643%

Tabla 22: Resultados set de validación balanceado (modelos bajo Python).

Los resultados, usando técnicas de balanceo, muestran diferencias entre las medidas de puntuación. Si tomamos las métricas de Accuracy y Balanced Accuracy se pueden ver grandes diferencias ya que estas medidas de puntuación están enfocadas a que, si los datos están balanceados o no, por eso, al revisar el Balanced Accuracy (59,418%) se ve una disminución importante en comparación al Accuracy (84,802%) que es más preciso

a la hora de analizar información balanceada, es decir, una diferencia importante de 25,384%.

6.4.2.2 Análisis de red Bayesiana (modelos Python)

Las redes Bayesianas causales son construidas a partir de modelos de probabilidad, por lo tanto, las aristas que conectan a dos nodos no deberían tener un significado más allá de la relación de dependencia entre ellos, es decir, una relación causa efecto, haciéndose conocida como modelo de causalidad. Dicho esto, la red Bayesiana a analizar es escogida entre los experimentos realizados, siendo el criterio de selección utilizado la que posea la mejor métrica de exactitud (accuracy) para datos balanceados y la métrica de exactitud balanceada (balanced accuracy) para datos desbalanceados. Para el caso de los datos desbalanceados la mejor métrica corresponde a un 63,41% y para los datos balanceados, la mejor métrica la genera bdeu con un 85,23%.

Cabe mencionar que cada experimento genera 5 DAG y el seleccionado para análisis se obtiene por aproximación al promedio anterior (85,23%). La siguiente tabla muestra los resultados de la validación cruzada de bdeu con datos balanceados.

# Partición	Accuracy
1	82,98%
2	85,11%
3	82,80%
4	87,10%
5	88,17%
Promedio	85,23%

Tabla 23: Particiones para BDEU con datos balanceados con foco en la métrica Accuracy.

El siguiente DAG corresponde a la partición 2.

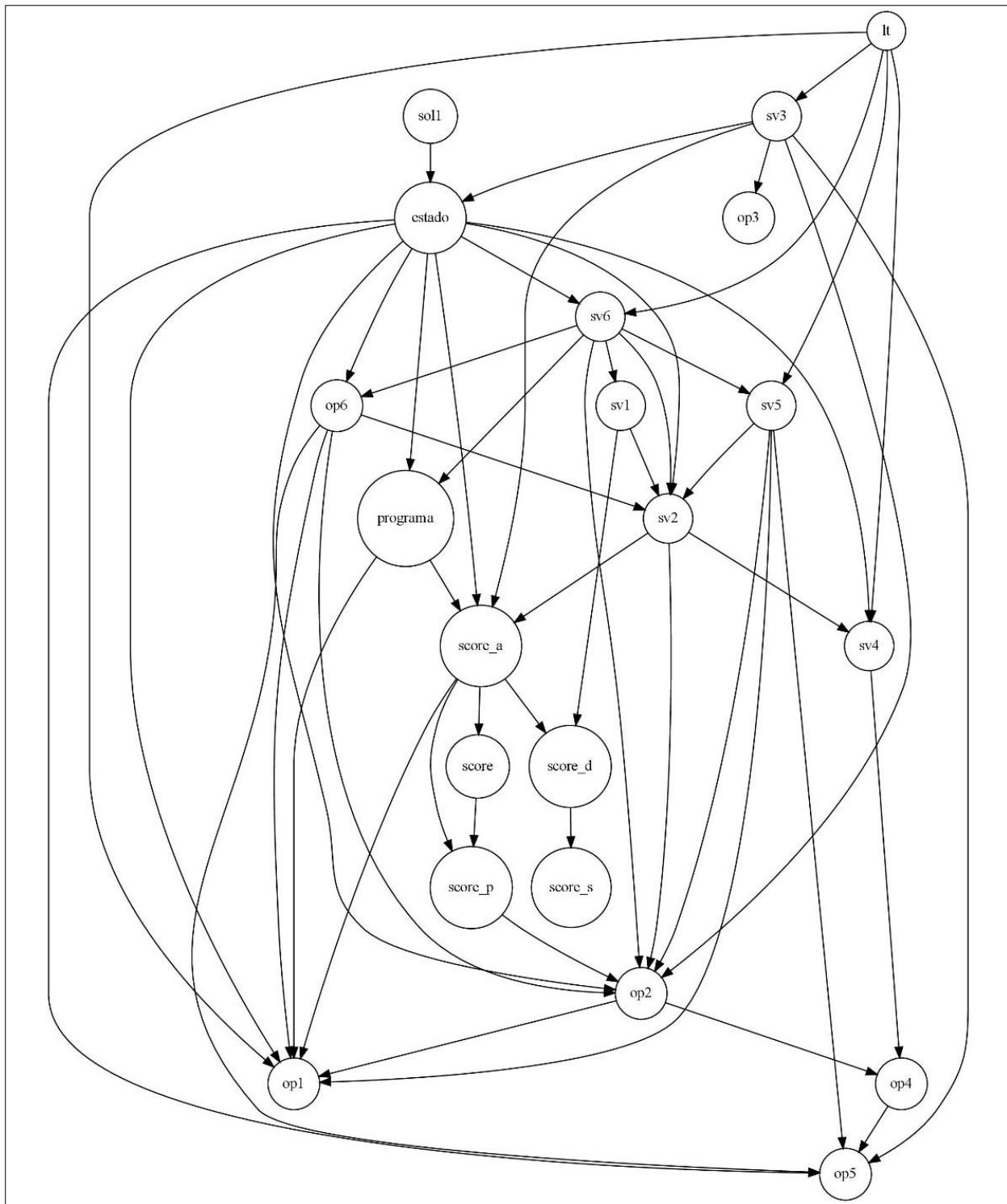


Figura 20: DAG - Bdeu - Balanceado - Partición 2.

El DAG de la figura 27 permite revisar la causalidad de la variable **estado**, que determina si un estudiante aprueba o reprueba el curso de programación. En el grafo se puede apreciar que la variable **estado** está directamente influenciada por la variable **sol1** (nota primera solemne) y la variable **sv3** (uso de más instrucciones permitidas en nivel 3 del juego de diagnóstico). En esta relación de causalidad, según lo observado en las tablas de probabilidades condicionales¹, estas permiten determinar que, si se usaron más instrucciones permitidas en el nivel 3 del juego de diagnóstico y la nota de la primera solemne no fue suficiente, existe una probabilidad media-alta de reprobación del curso.

En contra partida, la variable **estado** se ve influenciada, por 9 variables que corresponden a:

- **op1**: Encontró solución óptima en el nivel 1 de la prueba de diagnóstico
- **op2**: Encontró solución óptima en el nivel 2 de la prueba de diagnóstico
- **op5**: Encontró solución óptima en el nivel 5 de la prueba de diagnóstico
- **op6**: Encontró solución óptima en el nivel 6 de la prueba de diagnóstico
- **sv2**: Uso más instrucciones permitidas en nivel 2 del juego de diagnóstico
- **sv4**: Uso más instrucciones permitidas en nivel 4 del juego de diagnóstico
- **sv6**: Uso más instrucciones permitidas en nivel 6 del juego de diagnóstico
- **score_a**: Puntaje de abstracción en la prueba de diagnóstico
- **programa**: Carrera de origen del estudiante

En la observación se puede notar que existen varias relaciones que pueden ser transitivas, por ejemplo, una de ellas es:

$$(\text{estado} \rightarrow \text{op2}) \text{ y } (\text{estado} \rightarrow \text{op6} \rightarrow \text{op2})$$

Esto podría indicar que la relación $(\text{estado} \rightarrow \text{op6} \rightarrow \text{op2})$ sería redundante.

¹ Para conocer más sobre las probabilidades condicionales de los experimentos, estos pueden ser encontrados en el sitio web de GitHub donde se han dejado todos los resultados de manera pública. Además, se pueden encontrar todos los DAG obtenidos, junto las inferencias y el código fuente completo. <https://github.com/rtarbes/tesisBN>

Lo que están indicando las 9 relaciones listadas anteriormente, es que la condición de aprobación o reprobación del curso dependerá de la combinatoria de resultados al encontrar las soluciones optimas (op1, op2, op5 y op6) junto con haber usado más instrucciones (sv2, sv4 y sv6) en el juego de diagnóstico más el puntaje de abstracción obtenido en la prueba de diagnóstico. La variable **programa** no aporta información dado que es transitiva de la variable **score_a**, pero si puede entregar una información directa como variable independiente.

6.4.3 Modelo bajo R

En la búsqueda de un mejor modelo de red Bayesiana, se realizó una variedad de experimentos que entregaba conocimiento de cómo la red se conformaba, pero en determinado punto de la experimentación empezaron a aparecer dificultades técnicas que no permitían seguir experimentando, una de las principales fue la aplicación de restricciones de arcos en el aprendizaje de estructuras. La dificultad de aplicar restricciones de arcos fue detectada en el uso de la librería de bnlearn para Python, ya que esta, a pesar de que acepta la aplicación de listas negras y listas blancas, no son aplicables a arcos, sino que las usa para eliminar variables, entonces los anteriores llevó la investigación al uso de la librería bnlearn de R, por lo que la programación subió de nivel al incorporar R sobre Python por medio de la librería rpy2.

Por lo anterior, todos los experimentos ejecutados en el modelo bajo Python del punto 6.4.2, fueron repetidos y ejecutados con R sobre Python. La primera serie de experimentos ejecutados fueron los siguientes:

#	¿Balanceado?	Tipo de datos	Medida de puntuación	Estado
1	NO	Discretos	AIC	OK
2	NO	Discretos	BIC	OK
3	NO	Discretos	LOGLIK	No Ejecutado
4	NO	Discretos y Continuos	AIC-CG	OK
5	NO	Discretos y Continuos	BIC-CG	OK
6	NO	Discretos y Continuos	LOGLIK-CG	OK
7	SI	Discretos	AIC	OK
8	SI	Discretos	BIC	OK
9	SI	Discretos	LOGLIK	No Ejecutado
10	SI	Discretos y Continuos	AIC-CG	OK
11	SI	Discretos y Continuos	BIC-CG	OK
12	SI	Discretos y Continuos	LOGLIK-CG	No Ejecutado

Tabla 24: Listado de experimentos ejecutados utilizando R sobre Python,

En la tabla 24 se puede ver que existen 3 experimentos que no fueron ejecutados y esto fue debido a la capacidad de la memoria disponible al momento de ejecutar los experimentos no fue la apropiada para poder llevar a término esos casos entregando un error (Error: cons memory exhausted (limit reached?)).

Posteriormente, se ejecutó una segunda serie de experimentos para probar la selección de variables y restricciones de arcos, pero con la salvedad que solo fueron aquellos con un mejor rendimiento de acuerdo con un criterio de selección. En el caso de los experimentos de selección de variables se aplicaron en múltiplos de 5 con límite en 20 ya que la cantidad de variables analizadas llega a 24.

Cabe destacar que los tiempos de ejecución, utilizando bnlearn de R sobre Python, son muy buenos, es decir, cada ejecución toma aproximadamente de entre 3 a 5 minutos, a excepción de aquellos experimentos que no se ejecutaron debido al error en la capacidad de la memoria.

6.4.3.1 Resultado de experimentos de modelo bajo R

La siguiente tabla muestra un resumen del promedio de las métricas obtenidas del **set de validación (test)** para todos los experimentos ejecutados.

#		Tipo de datos	Fx puntuación	Accuracy	Balanced Accuracy	Precision Score	Recall Score	ROC AUC	Class Ratio
1	NO	Discreto	AIC	84,994%	64,480%	83,815%	84,994%	66,416%	10,492%
2	NO	Discreto	BIC	84,999%	60,186%	85,937%	84,999%	59,261%	10,492%
3	NO	Discreto y Continuo	AIC-CG	86,511%	62,989%	87,974%	86,511%	60,100%	10,492%
4	NO	Discreto y Continuo	BIC-CG	88,865%	69,350%	93,040%	88,865%	59,543%	10,492%
5	NO	Discreto y Continuo	LOGLIK-CG	49,007%	55,328%	69,848%	49,007%	61,716%	10,492%
6	SI	Discreto	AIC	70,663%	55,207%	66,865%	70,663%	60,180%	10,492%
7	SI	Discreto	BIC	71,293%	54,919%	67,138%	71,293%	58,986%	10,492%
8	SI	Discreto y Continuo	AIC-CG	89,732%	71,272%	91,944%	89,732%	66,408%	10,492%
9	SI	Discreto y Continuo	BIC-CG	78,591%	53,853%	75,132%	78,591%	55,678%	10,492%

Tabla 25: Resultado de métricas para el set de validación para 9 experimentos ejecutados.

Para la ejecución de la segunda serie de experimentos se realizó una selección de dos experimentos candidatos para la aplicación del método de selección de variables y para el uso de restricciones de arcos. Los candidatos fueron elegidos observando las métricas resultantes, principalmente la métrica utilizada para la selección de los mejores es la curva ROC que permite medir el rendimiento de clasificación en base a la tasa de verdaderos positivos y la tasa de falsos positivos.

Entonces, bajo el criterio explicado, se elige el experimento número 1 con un ROC de un 66,416% del grupo de experimentos no balanceados y el número 8 con un ROC de 66,408% del grupo de experimentos balanceados.

Los experimentos seleccionados fueron sometidos a una nueva ronda de experimentación aplicando el método de selección de variables en rangos de múltiples de 5 con límite de 20. La siguiente tabla muestra los resultados de estos experimentos.

#		# de var.	Tipo de datos	Fx puntuac.	Accuracy	Balanced Accuracy	Precision Score	Recall Score	ROC AUC	Class Ratio
1	NO	5	Discreto	AIC	90,144%	74,508%	92,190%	90,144%	66,535%	10,492%
2	NO	10	Discreto	AIC	87,149%	65,598%	87,519%	87,149%	65,743%	10,492%
3	NO	15	Discreto	AIC	86,497%	65,213%	86,531%	86,497%	65,380%	10,492%
4	NO	20	Discreto	AIC	86,500%	65,568%	86,361%	86,500%	66,375%	10,492%
5	SI	5	Discreto y Continuo	AIC-CG	87,362%	66,634%	87,771%	87,362%	65,859%	10,492%
6	SI	10	Discreto y Continuo	AIC-CG	86,092%	64,721%	85,461%	86,092%	66,906%	10,492%
7	SI	15	Discreto y Continuo	AIC-CG	85,010%	62,260%	84,451%	85,010%	63,662%	10,492%
8	SI	20	Discreto y Continuo	AIC-CG	85,866%	61,686%	86,570%	85,866%	60,509%	10,492%

Tabla 26: Listado de experimentos ejecutados por método de selección de variables.

He de recordar que esta serie de experimentos derivan de dos anteriormente expuestos cuyas métricas, sin aplicar el método de selección de variables, son:

#		Tipo de datos	Fx puntuación	Accuracy	Balanced Accuracy	Precision Score	Recall Score	ROC AUC	Class Ratio
1	NO	Discreto	AIC	84,994%	64,480%	83,815%	84,994%	66,416%	10,492%
8	SI	Discreto y Continuo	AIC-CG	89,732%	71,272%	91,944%	89,732%	66,408%	10,492%

Tabla 27: Métricas de los experimentos seleccionados para la aplicación del método de selección de variables.

En resumen, si comparamos las métricas originales con las métricas después de aplicar el método de selección de variables tenemos la siguiente tabla comparativa para el experimento 1 con datos desbalanceados y discretos con medida de puntuación AIC.

Métrica	Original	5	10	15	20
Accuracy	84,994%	90,144%	87,149%	86,497%	86,500%
Balanced Accuracy	64,480%	74,508%	65,598%	65,213%	65,568%
Precision Score	83,815%	92,190%	87,519%	86,531%	86,361%
Recall Score	84,994%	90,144%	87,149%	86,497%	86,500%
ROC AUC	66,416%	66,535%	65,743%	65,380%	66,375%
Class Ratio	10,492%	10,492%	10,492%	10,492%	10,492%

Tabla 28: Tabla comparativa de experimento con datos desbalanceados y discretos con medida de puntuación AIC con respecto a la aplicación del método selección de variables (5, 10, 15 y 20 variables).

Se observa una buena mejoría de la métrica Balanced Accuracy del modelo original al aplicar el método de selección con 5 variables y una pequeña mejora en los otros experimentos.

La siguiente tabla muestra la comparativa del segundo modelo seleccionado y que corresponde al experimento 8 con datos balanceados, datos discretos y continuos usando la medida de puntuación AIC – CG.

Métrica	Original	5	10	15	20
Accuracy	89,732%	87,362%	86,092%	85,010%	85,866%
Balanced Accuracy	71,272%	66,634%	64,721%	62,260%	61,686%
Precision Score	91,944%	87,771%	85,461%	84,451%	86,570%
Recall Score	89,732%	87,362%	86,092%	85,010%	85,866%
ROC AUC	66,408%	65,859%	66,906%	63,662%	60,509%
Class Ratio	10,492%	10,492%	10,492%	10,492%	10,492%

Tabla 29: Tabla comparativa de experimento con datos balanceados, discretos y continuos con medida de puntuación AIC-CG con respecto a la aplicación del método de selección de variables (5, 10 ,15, 20 variables).

Es este experimento no se observa, en la métrica Accuracy, una mejoría del modelo con la aplicación del método de selección de variables y el único valor observable que aumento fue la métrica ROC con 10 variables que aumentó de 66,408% a 66,906%, es decir, tuvo una variación positiva de 0,498%. En resumen, cuando se usan datos balanceados la aplicación de métodos de selección de variables no genera mejorías al modelo.

Para finalizar se ejecutaron dos experimentos más aplicando restricciones de arcos para ambos modelos seleccionados (experimento 1 y 8). Para poder llevar a cabo estos casos

se analizaron las métricas de ambos modelos a nivel del número de particiones aplicado a todos los experimentos ejecutados. Las siguientes tablas permiten observar las métricas abiertas por partición de cada modelo.

# Partición	Accuracy	Balanced Accuracy	Precision Score	Recall Score	ROC AUC	Class Ratio
1	87,234%	64,680%	88,597%	87,234%	62,024%	10,638%
2	90,426%	74,837%	90,907%	90,426%	72,619%	10,638%
3	87,097%	68,897%	85,791%	87,097%	75,181%	10,753%
4	79,570%	55,513%	76,584%	79,570%	57,771%	10,753%
5	80,645%	58,475%	77,197%	80,645%	64,484%	9,677%
Promedio	84,994%	64,480%	83,815%	84,994%	66,416%	10,492%
Mediana		64,680%				

Tabla 30: Tabla de métricas por partición del experimento con datos desbalanceados y discretos usando la medida de puntuación AIC.

Para seleccionar la estructura aprendida (DAG) se utilizó la métrica Balanced Accuracy que permite evaluar que tan bueno es un clasificador sobre todo cuando las clases están desequilibradas. En este caso particular la partición seleccionada se obtiene, de acuerdo con el promedio y la mediana, que en este caso corresponde a 64,480% y 64,680% respectivamente. Con estos datos se elige la partición más cercana correspondiente a la partición 1 con un 64,680%

La siguiente es la tabla del experimento 8 con datos balanceados, discretos y continuos con la medida de puntuación AIC – CG.

# Partición	Accuracy	Balanced Accuracy	Precision Score	Recall Score	ROC AUC	Class Ratio
1	87,234%	62,121%	90,436%	87,234%	57,619%	10,638%
2	87,234%	57,500%	92,751%	87,234%	53,214%	10,638%
3	89,247%	71,988%	89,247%	89,247%	71,988%	10,753%
4	91,398%	77,590%	91,398%	91,398%	77,590%	10,753%
5	93,548%	87,159%	95,887%	93,548%	71,627%	9,677%
Promedio	89,732%	71,272%	91,944%	89,732%	66,408%	10,492%
Mediana	89,247%					

Tabla 31: Tabla de métricas por partición del experimento con datos balanceados, discretos y continuos con la medida de puntuación AIC-CG.

Para seleccionar la estructura aprendida (DAG) se utilizó la métrica Accuracy que permite evaluar el modelo cuando los datos están balanceados. En este caso la partición seleccionada se obtiene, de acuerdo con el promedio y la mediana, que en este caso corresponde a 89,732% y 89,247% respectivamente. Con estos datos se escoge la partición más cercana correspondiente a la partición 3 con 89,274%.

Una vez determinados los DAG, se analizan para determinar cuáles son los arcos que serán aplicados como lista negra y como lista blanca en cada caso. A continuación, se muestran los DAG de las particiones seleccionadas.

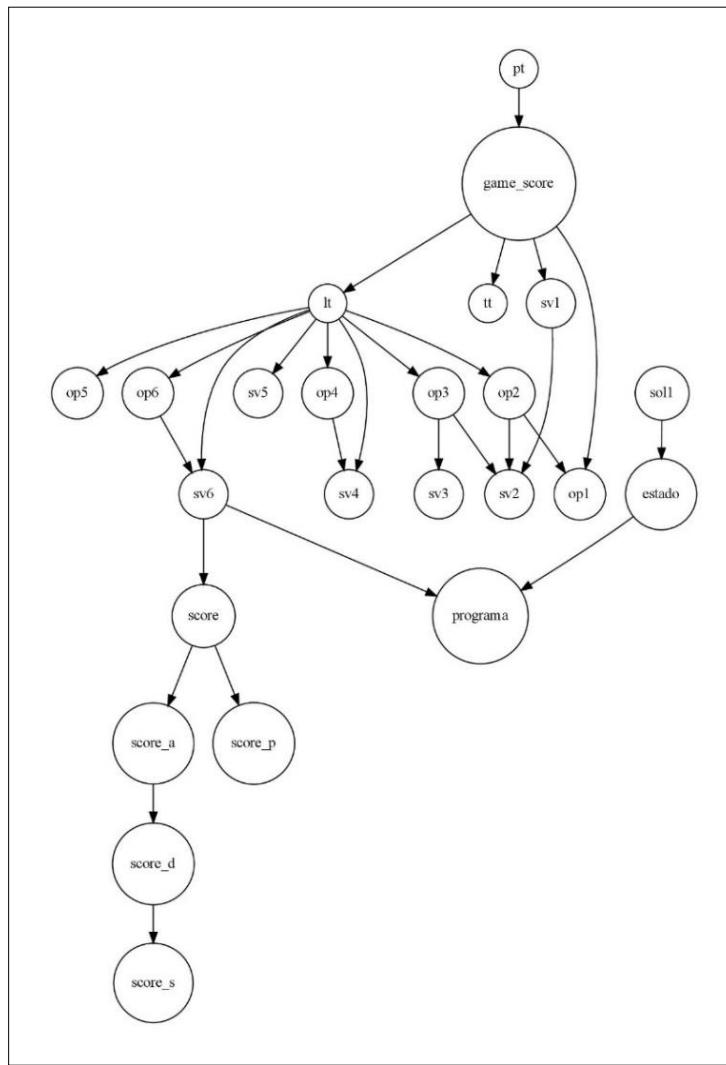


Figura 21: Red con datos desbalanceados y discreto usando medida de puntuación AIC.

Para el caso de la figura 21, se observa que existen relaciones que rompen la temporalidad de la información. Lo anterior, debido a que en los datos utilizados existen 6 niveles de un juego de diagnóstico que son secuenciales donde la variable **It** (largo total acumulado de las soluciones) no es resuelta hasta que los niveles están completados. Adicionalmente, existe otro caso donde una solución óptima del nivel 3 (**op3**) es independiente y temporalmente posterior a determinar si el alumno uso más de las instrucciones permitidas en el nivel 2 (**sv2**), por lo tanto, se aplicaría una lista negra con los siguientes arcos:

- **lt → sv4**
- **lt → sv6**
- **op3 → sv2**

También se detecta que se puede invertir la arista entre los nodos **op2** y **op1** (**op2 → op1**) por temporalidad ya que encontrar la solución óptima del nivel 1 ocurre primero que el nivel 2. Otro caso es el arco **estado → programa** dado que por conocimiento experto el programa de estudio de un alumno tiene una incidencia sobre el estado de aprobación o reprobación del curso. Entonces, para esos casos se aplica una lista blanca con los siguientes arcos:

- **op1 → op2**
- **programa → estado**

A continuación, se muestra cómo queda representada la red después de aplicar las restricciones. La red que se muestra es la que mejor puntuación tiene de la métrica balanced accuracy (74,837%) para las 5 particiones.

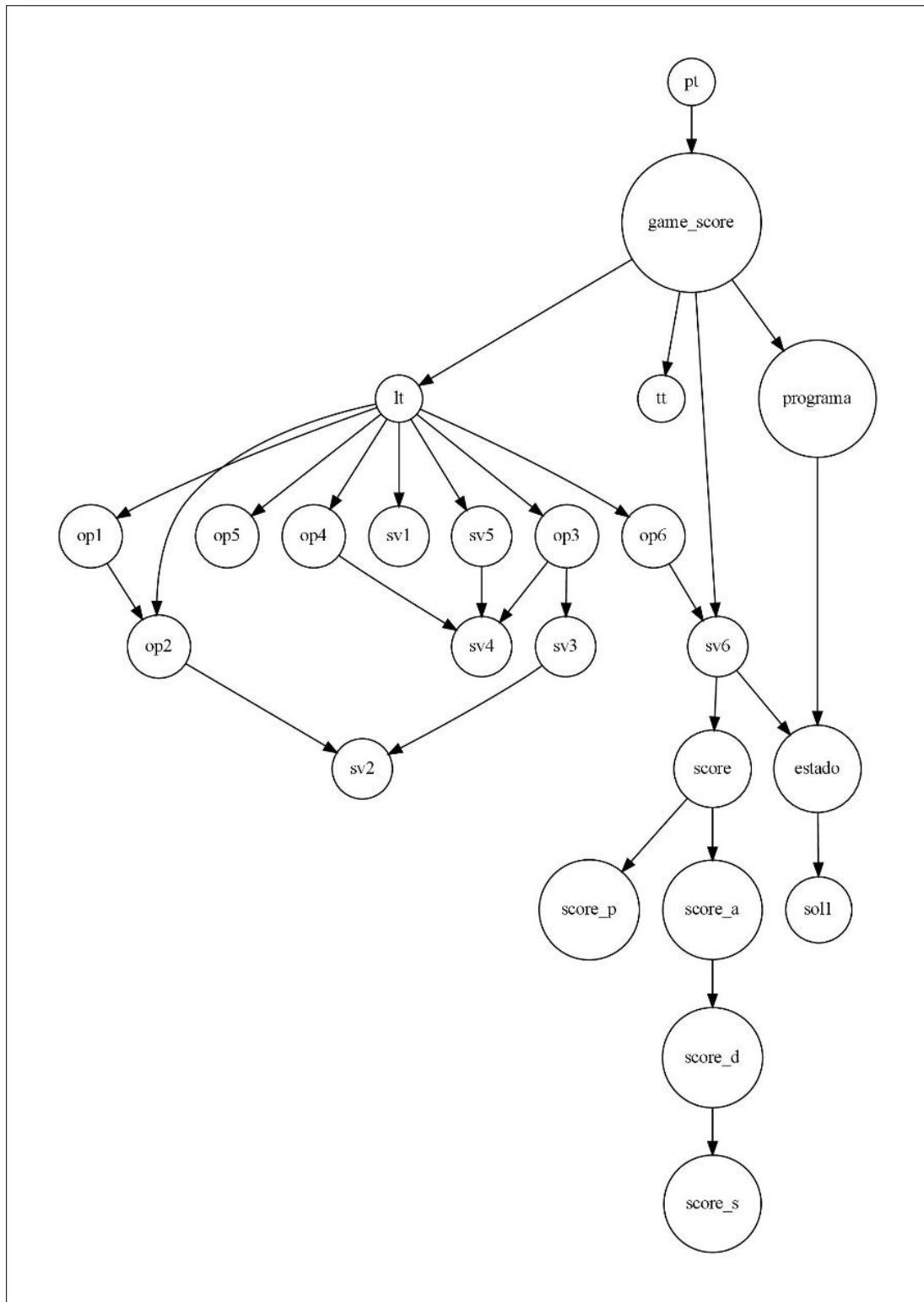


Figura 22: Red con datos desbalanceados y discreto usando medida de puntuación AIC después de aplicar restricciones.

A continuación, se presenta la red con datos balanceados, discretos y continuos usando medida de puntuación AIC – CG.

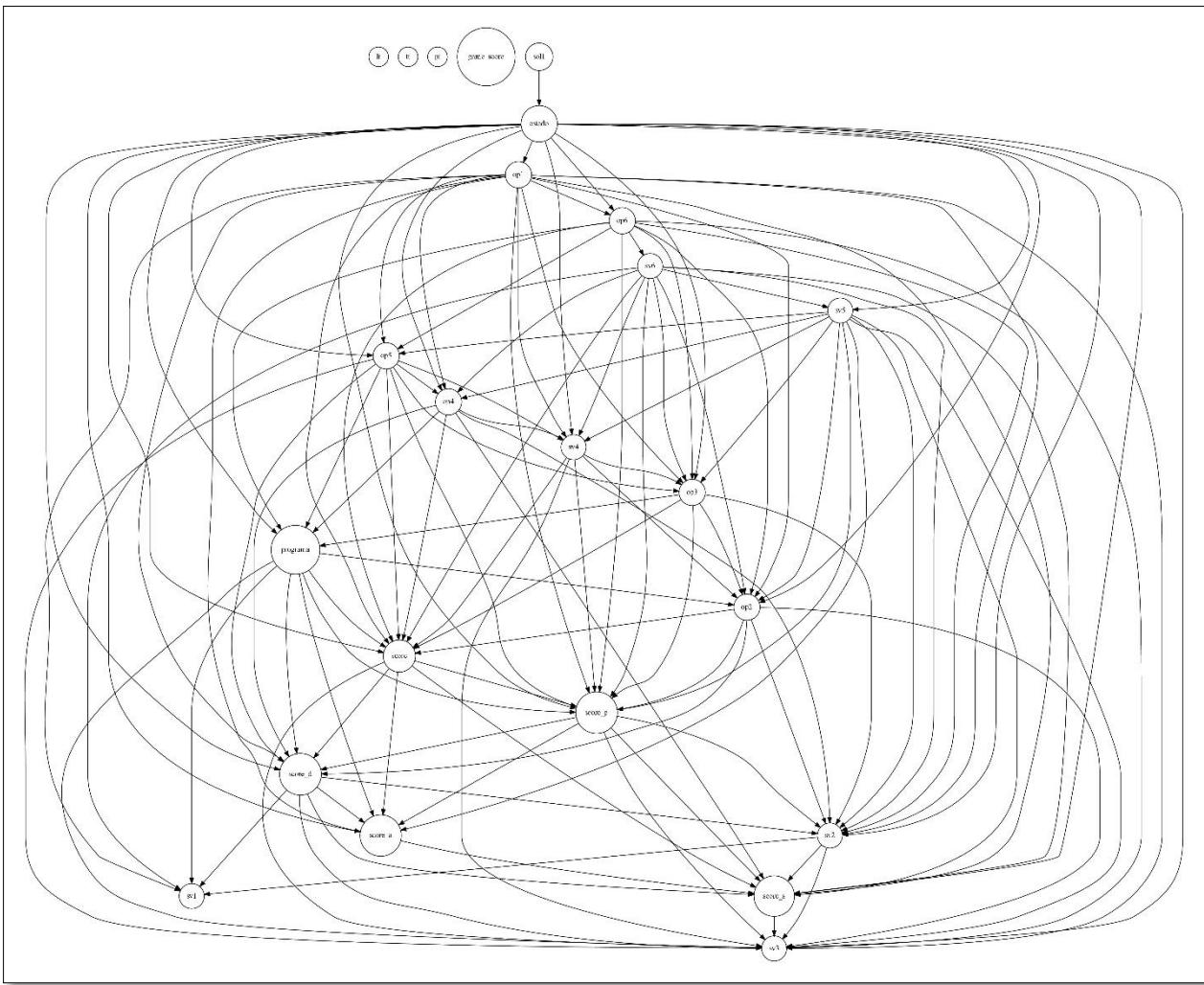


Figura 23: Red con datos balanceados, discretos y continuos usando medida de puntuación AIC-CG.

Para el caso de la figura 23, el aprendizaje de estructura aplicado por la medida de puntuación AIC – CG genera muchas relaciones, por lo que el análisis para encontrar las restricciones también se basa en la temporalidad de la información encontrando los siguientes arcos a aplicar en lista negra:

- $op1 \rightarrow op3$
 - $op1 \rightarrow op4$
 - $op1 \rightarrow op5$

- **op1 → op6**
- **op1 → sv4**
- **op2 → sv3**
- **op4 → sv2**
- **op5 → op3**
- **op5 → sv3**
- **op6 → op2**
- **op6 → op3**
- **op6 → sv2**
- **op6 → sv3**
- **sv4 → op2**
- **sv4 → op3**
- **sv5 → op2**
- **sv5 → op3**
- **sv6 → op2**
- **sv6 → op3**
- **sv6 → op4**

Dado que el aprendizaje de la estructura de este experimento no encontró relación de la variable **It** (largo total acumulado de las soluciones) con nada, y dado el conocimiento adquirido de la información, se determina que esta debe tener una relación con las variables que indican si el alumno encontró una solución óptima en cada nivel del juego de diagnóstico (**op1, op2, op3, op4, op5, op6**). Adicionalmente, la variable **game_score** también quedó sin ninguna relación, por lo tanto, también se aplica el conocimiento experto que indica que esa variable está relacionada con la variable **pt** (total de pruebas de los 6 niveles) se determinan los siguientes arcos como lista blanca:

- **lt → op1**
- **lt → op2**
- **lt → op3**
- **lt → op4**
- **lt → op5**

- **lt → op6**
- **pt → game_score**

A continuación, se muestra cómo queda representada la red después de aplicar las restricciones. La red que se muestra es la que mejor puntuación tiene de la métrica Accuracy (92,473%) para las 5 particiones.

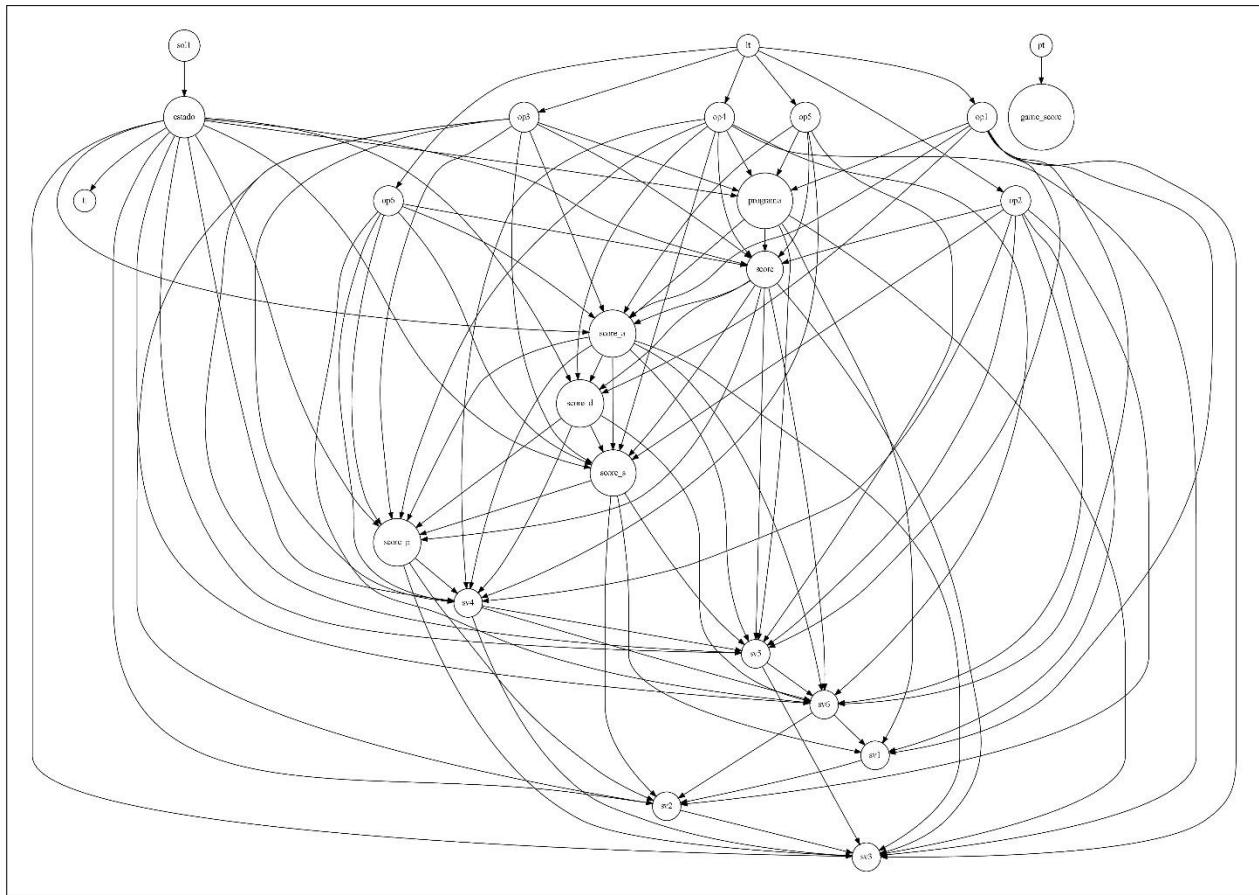


Figura 24: Red con datos balanceados, discretos y continuos usando medida de puntuación AIC-CG después de aplicar restricciones.

De ambos experimentos con restricciones se concluyeron las siguientes métricas:

#		Tipo de datos	Fx puntuación	Accuracy	Balanced Accuracy	Precision Score	Recall Score	ROC AUC	Class Ratio
1	SI	Discreto y Continuos	AIC-CG	88,868%	70,607%	90,928%	88,868%	64,056%	10,492%
2	NO	Discreto	AIC	84,994%	64,480%	83,815%	84,994%	66,416%	10,492%

Tabla 32: Resultado de métricas de los experimentos seleccionados una vez aplicada las restricciones de arcos.

A continuación, se hace una comparativa de los experimentos seleccionados junto a la aplicación de restricción de arcos. La siguiente tabla compara ambos experimentos con datos balanceados, discretos y continuos usando la medida de puntuación AIC-CG

Métrica	Original	Restricción
Accuracy	89,732%	88,868%
Balanced Accuracy	71,272%	70,607%
Precision Score	91,944%	90,928%
Recall Score	89,732%	88,868%
ROC AUC	66,408%	64,056%
Class Ratio	10,492%	10,492%

Tabla 33: Tabla comparativa de experimento con datos balanceados, discretos y continuos usando medida de puntuación AIC-CG con respecto a la aplicación de restricción de arcos.

Al comparar los experimentos no muestra mejoría al aplicar las restricciones, disminuyendo el Accuracy en 0.864%.

Para el caso del experimento con datos desbalanceados y discretos usando la medida de puntuación AIC, la siguiente tabla muestra la comparativa de resultados.

Métrica	Original	Restricción
Accuracy	84,994%	84,994%
Balanced Accuracy	64,480%	64,480%
Precision Score	83,815%	83,815%
Recall Score	84,994%	84,994%
ROC AUC	66,416%	66,416%
Class Ratio	10,492%	10,492%

Tabla 34: Tabla comparativa de experimento con datos desbalanceados y discretos usando medida de puntuación AIC con respecto a la aplicación de restricción de arcos.

El comportamiento para ambos experimentos no generó una variación significativa lo que se puede reforzar analizando el resultado de la métrica Balanced Accuracy por cada partición:

# Partición	Original	Restricción
1	0,64680233	0,64680233
2	0,74836601	0,67784553
3	0,68896926	0,71410256
4	0,55512821	0,60488246
5	0,58475232	0,58475232
promedio	0,64480	0,64568
mediana	0,64680	0,64680

Tabla 35: Tabla comparativa de experimento con datos desbalanceados y discretos usando medida de puntuación AIC por cada partición con respecto a la aplicación de restricción de arcos.

En el detalle por partición se puede ver un aumento no muy significativo de 0.0873% respecto al promedio de las particiones.

En resumen, en este apartado se revisaron los distintos experimentos ejecutados, por medio del programa construido, utilizando lenguaje R sobre Python. A continuación, se lista un resumen de los experimentos seleccionados junto a la selección de variables y la aplicación de restricciones de arcos.

#	Tipo	Tipo de datos	Fx puntuación	Accuracy	Balanced Accuracy	Class Ratio	Precision Score	Recall Score	ROC AUC
1	Original	Discretos	AIC	84,994%	64,480%	10,492%	83,815%	84,994%	66,416%
2	5 var.	Discretos	AIC	90,144%	74,508%	10,492%	92,190%	90,144%	66,535%
3	10 var.	Discretos	AIC	87,149%	65,598%	10,492%	87,519%	87,149%	65,743%
4	15 var.	Discretos	AIC	86,497%	65,213%	10,492%	86,531%	86,497%	65,380%
5	20 var.	Discretos	AIC	86,500%	65,568%	10,492%	86,361%	86,500%	66,375%
6	Restric.	Discretos	AIC	84,994%	64,480%	10,492%	83,815%	84,994%	66,416%

Tabla 36: Comparativa de experimentos con datos desbalanceados y discretos usando medida de puntuación AIC.

Según la información que entrega la tabla comparativa el mejor Balanced Accuracy lo está entregando el experimento con selección de 5 variables con un 74,508%, seguido de la selección de 10 variables con un 65,598%.

Para la tabla comparativa con datos balanceados, discretos y continuos usando AIC-CG como medida de puntuación se tiene que:

#	Tipo	Tipo de datos	Fx puntuación	Accuracy	Balanced Accuracy	Class Ratio	Precision Score	Recall Score	ROC AUC
1	Original	Discretos y continuos	AIC-CG	86,511%	62,989%	10,492%	87,974%	86,511%	60,100%
2	5 var.	Discretos y continuos	AIC-CG	87,362%	66,634%	10,492%	87,771%	87,362%	65,859%
3	10 var.	Discretos y continuos	AIC-CG	86,092%	64,721%	10,492%	85,461%	86,092%	66,906%
4	15 var.	Discretos y continuos	AIC-CG	85,010%	62,260%	10,492%	84,451%	85,010%	63,662%
5	20 var.	Discretos y continuos	AIC-CG	85,866%	61,686%	10,492%	86,570%	85,866%	60,509%
6	Restric.	Discretos y continuos	AIC-CG	88,868%	70,607%	10,492%	90,928%	88,868%	64,056%

Tabla 37: Comparativa de experimentos con datos balanceados, discretos y continuos usando medida de puntuación A/C-CG.

Para este grupo de experimentos se observa que el mejor Accuracy lo entrega el experimento con restricciones con un 88,868% seguido de la selección de 5 variables con un 87,362%.

Para finalizar, en el siguiente apartado se realizará un análisis de las redes mejor puntuadas en este resumen.

6.4.3.2 Análisis de redes Bayesianas (modelo R)

Dado que las redes Bayesianas son construidas a partir de modelos de probabilidad se realizará un análisis de las dos mejores redes Bayesianas determinadas por las métricas discutidas en el apartado anterior sobre los resultados de experimentos de modelo bajo R.

Los modelos para analizar corresponden a los siguientes:

- i. Modelo con datos desbalanceados y discretos usando medida de puntuación AIC usando método de selección de 5 variables.
- ii. Modelo con datos balanceados, discretos y continuos usando medida de puntuación AIC-CG con restricción de arcos

Dado que los modelos fueron seleccionados en base al promedio de las métricas, la selección de la red Bayesiana para ser analizada se escogió en base al criterio de la

mediana de las 5 particiones. Las siguientes tablas muestran el detalle por partición de cada experimento:

# Partición	Accuracy	Balanced Accuracy	Precision Score	Recall Score	ROC AUC	Class Ratio
1	89,362%	71,023%	92,188%	89,362%	63,214%	10,638%
2	93,617%	88,826%	95,694%	93,617%	74,405%	10,638%
3	89,247%	71,471%	90,434%	89,247%	67,590%	10,753%
4	87,097%	64,632%	88,473%	87,097%	61,988%	10,753%
5	91,398%	76,591%	94,163%	91,398%	65,476%	9,677%
Promedio	90,144%	74,508%	92,190%	90,144%	66,535%	10,492%
Mediana	89,362%	71,471%	92,188%	89,362%	65,476%	10,638%

Tabla 38: Particiones del experimento con datos desbalanceados y discretos con medida de puntuación AIC con método de selección de 5 variables.

De acuerdo con la mediana de la tabla 38, la partición más cercana, con respecto a la métrica Balanced Accuracy corresponde a la partición 3, por lo tanto, la red Bayesiana de esta partición será utilizada para ser analizada.

# Partición	Accuracy	Balanced Accuracy	Precision Score	Recall Score	ROC AUC	Class Ratio
1	88,298%	65,506%	92,363%	88,298%	58,214%	10,638%
2	88,298%	67,406%	90,380%	88,298%	62,619%	10,638%
3	86,022%	64,523%	85,467%	86,022%	65,783%	10,753%
4	89,247%	71,471%	90,434%	89,247%	67,590%	10,753%
5	92,473%	84,129%	95,993%	92,473%	66,071%	9,677%
Promedio	88,868%	70,607%	90,928%	88,868%	64,056%	10,492%
Mediana	88,298%	67,406%	90,434%	88,298%	65,783%	10,638%

Tabla 39: Particiones del experimento con datos balanceados, discretos y continuos con medida de puntuación AIC-CG con restricción de arcos.

De acuerdo con la mediana de la tabla 39, la partición más cercana, con respecto a la métrica Accuracy corresponde a la partición 1 y 2, por lo tanto, se escogerá la red Bayesiana que presente todas sus variables conectadas y esa corresponde a la partición 1 para ser utilizada en el análisis.

Ahora que se tiene seleccionada las redes Bayesiana a ser analizadas, la siguiente figura muestra la red correspondiente a la partición 3 del experimento con datos desbalanceados y discretos con medida de puntuación AIC y con método de selección de 5 variables.

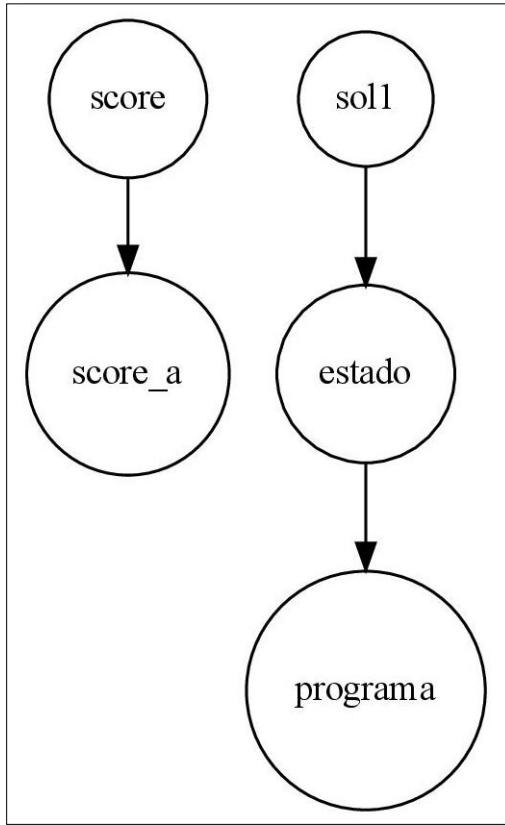


Figura 25: Red con datos desbalanceados y discretos con medida de puntuación AIC y con método de selección de 5 variables.

La red Bayesiana de la figura 25 representa a las 5 variables seleccionadas por el método de selección de variables. Estas 5 variables son las que mejor puntuación obtuvieron al ejecutar método de selección. Aquí se puede apreciar la directa relación que poseen las variables sol1, estado y programa indicando que la nota de la primera solemne (sol1) tiene una fuerte influencia en el estado de aprobación o reprobación del curso y el resultado del curso está directamente relacionado a la carrera del estudiante (programa), esto último indica que, dependiendo de la carrera del alumno, tendrá o no éxito en la aprobación del curso. Por su parte, el puntaje total de la prueba de diagnóstico (score) tiene una fuerte

influencia sobre el puntaje de abstracción (score_a), es decir, a mayor puntaje total mayor es el puntaje de abstracción, esto último de acuerdo con las tablas de probabilidad condicional.

La siguiente red Bayesiana corresponde a la partición 1 del experimento con datos balanceados, discretos y continuos con medida de puntuación AIC-CG y con restricción de arcos.

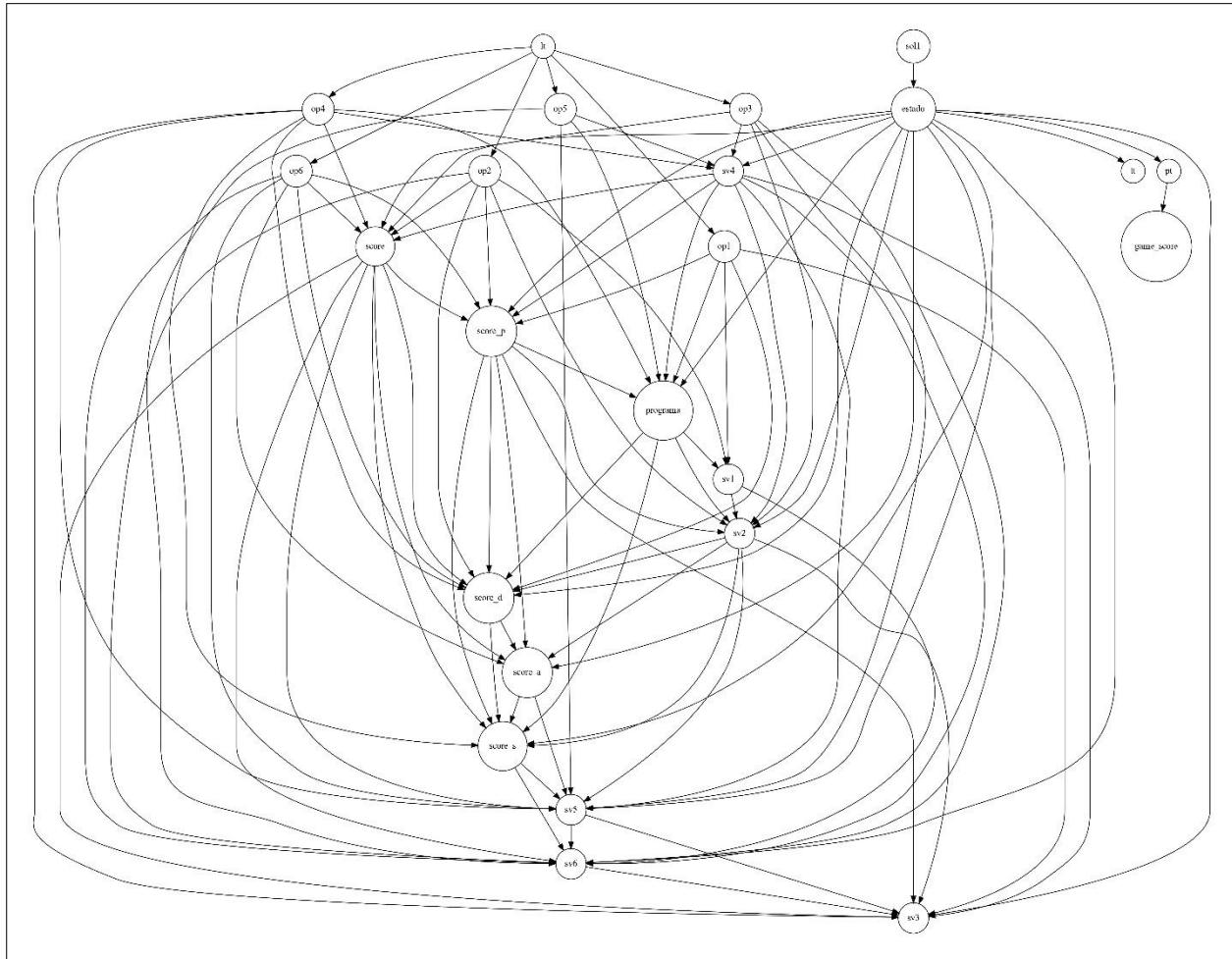


Figura 26: Red con datos balanceados, discretos y continuos con medida de puntuación AIC-CG y con restricción de arcos.

La red Bayesiana de la figura 26 es una red aprendida usando la medida de puntuación AIC-CG que corresponde al criterio de información de Akaike para redes Bayesianas

híbridas (variables categóricas y normales mixtas), esta medida de puntuación generó múltiples relaciones entre las variables.

A primera vista las variables que determinan si fueron encontrada las soluciones optimas en cada nivel del juego de diagnóstico (**op1, op2, op3, op4, op5, op6**) se encuentran en la parte superior de la red relacionadas con la variable que indica el largo total de las soluciones de cada nivel del juego (**lt**), es decir, que dependiendo del largo total acumulado de los niveles del juego, será determinante en identificar si el alumno encontró las soluciones optimas en cada nivel del juego.

También, se puede identificar que un alumno puede encontrar una solución óptima en cada nivel del juego de diagnóstico dependerá de la formación académica que este presente, es decir, la carrera de origen del alumno (**programa**).

En términos generales, si revisamos las variables de los puntajes (**score, score_p, score_d, score_a, score_s**), tenemos que un alumno, si obtiene buenos valores en la prueba de diagnóstico esto dependerá de que encuentre soluciones optimas en los todos los niveles del juego de diagnóstico. Si vemos en detalle, que un alumno obtiene un buen puntaje de reconocimiento de patrones (**score_p**) dependerá de que haya encontrado soluciones optimas en los niveles 1, 2, 6 y también dependerá si uso más de las instrucciones permitidas en el nivel 4 del juego de diagnóstico como también influye directamente con la variable **estado** del curso. Para el caso del puntaje de descomposición (**score_d**) dependerá de que el alumno haya encontrado soluciones optimas en el nivel 1, 2, 4, 6. También, se encuentra una relación con el uso de más instrucciones permitidas en el nivel 2 (**sv2**), es decir, si encontró más instrucciones permitidas en el nivel 2 del juego de diagnóstico, esta influye en el puntaje de descomposición como también en el **estado** del curso.

El obtener un puntaje de abstracción (**score_a**) tiene una baja incidencia con el puntaje de descomposición (**score_d**), es decir, si el puntaje de descomposición es bajo o alto no depende en gran medida de que el puntaje de abstracción sea bajo o alto. En el caso del estado del curso, el tener un buen o mal puntaje de abstracción tiene una dependencia media de que el alumno apruebe o repreuebe el curso. Este puntaje también depende que

el alumno haya encontrado una solución óptima en el nivel 6, pero el puntaje total de la prueba de diagnóstico tiene una fuerte incidencia en los puntajes individuales, como en este caso con el puntaje de abstracción.

Para el puntaje de algoritmos (**score_s**) existen relaciones con que el alumno encuentre una solución óptima en el nivel 4 (**op4**) del juego de diagnóstico, haya usado más de las instrucciones permitidas en el nivel 2 (**sv2**) y con los puntajes de abstracción (**score_a**), de reconocimiento de patrones (**score_p**), puntaje de descomposición (**score_d**) y sobre todo con relación a la carrera de origen del alumno (**programa**).

En resumen, si se pone el foco en el análisis de la variable **estado** del curso, este está directamente influenciado por la nota de la primera prueba solemne del curso (**sol1**). A su vez, el estado del curso dependerá distintos factores, como lo son el tiempo total (**tt**) que tomó realizar el juego de diagnóstico, que haya usado más de las instrucciones permitidas en los niveles 2, 3, 4, 5, 6 (**sv2, sv3, sv4, sv5, sv6**) y tenga buenos puntajes en algoritmos (**score_s**), abstracción (**score_a**), descomposición (**score_d**) y sobre todo en el puntaje total (**score**). En conclusión, que un alumno apruebe el curso dependerá que obtenga buenos puntajes en la prueba de diagnóstico y no haga uso de muchas instrucciones para encontrar las soluciones en el juego de diagnóstico.

6.4.4 Resultados de modelos bajo Weka

En el primer experimento con las variables del juego diagnóstico, al probar los algoritmos K2 y TabuSearch, se observa que sus métricas no son tan buenas como las de Hill Climber y Repeated Hill Climber, incluso sus grafos generados no parecen representar relaciones en coherencia con los experimentos en Python y R, por lo tanto, estos métodos son descartados. Lo mismo sucede con las modalidades locales de los algoritmos, y también son descartadas. Entonces, las métricas de los modelos generados con Hill Climber Global y Repeated Hill Climber Global son comparadas. Además, se utiliza el estimador MLE que resulta mejor que el Estimador Simple con Alfa 0.5. La tabla 40 muestra que no se observa alguna ventaja en rendimiento de uno sobre otro, y se selecciona el modelo de Hill Climber Global debido a que toma menos tiempo y procesamiento.

Algoritmo	Tasa VP	Tasa FP	Precisión	Recall	F-Measure	MCC	Área ROC	Área PRC
Hill Climber	0,869	0,860	0,814	0,869	0,838	0,016	0,496	0,811
Repeatead HC	0,871	0,878	0,807	0,871	0,836	-0,012	0,486	0,809

Tabla 40: Comparación de métricas de los dos algoritmos con mejor desempeño.

La matriz de confusión del modelo seleccionado muestra sólo 2 instancias de la clase “Reprobado” correctamente clasificadas. Entonces se balancea el modelo usando una penalización de 10, después de probar con varios valores. En la nueva matriz de confusión puede observarse que 16 instancias de la clase “Reprobado” fueron correctamente clasificadas, una mejora en comparación con el modelo desbalanceado. En la tabla 41 puede observarse, al comparar las métricas del modelo desbalanceado y balanceado, una mejora en el área ROC y en la tasa de falsos positivos, en desmedro de otras métricas.

Modelo	Tasa VP	Tasa FP	Precisión	Recall	F-Measure	MCC	Área ROC	Área PRC
Desbalanceado	0,869	0,860	0,814	0,869	0,838	0,016	0,496	0,811
Balanceado	0,660	0,628	0,816	0,660	0,721	0,021	0,507	0,813

Tabla 41: Comparación de métricas de modelo seleccionado desbalanceado y balanceado.

El grafo generado por el modelo balanceado presenta una fuerte concentración de arcos alrededor de la clase “estado”, en comparación con los otros nodos (ver figura 27), en total son 11 arcos, lo que puede significar la existencia de influencia directa de los resultados o métricas de algunos niveles del juego diagnóstico sobre la clase. En particular, existe relación con las variables “lt”, “pt” y “tt” que representan las métricas totales del juego, y esto puede significar que posiblemente el resultado final del juego tenga influencia sobre el estado de reprobación / aprobación del estudiante, a pesar de no existir una relación directa con la variable “game_score”. Los nodos aislados pueden reflejar la falta de influencia de los resultados o métricas parciales que representan.

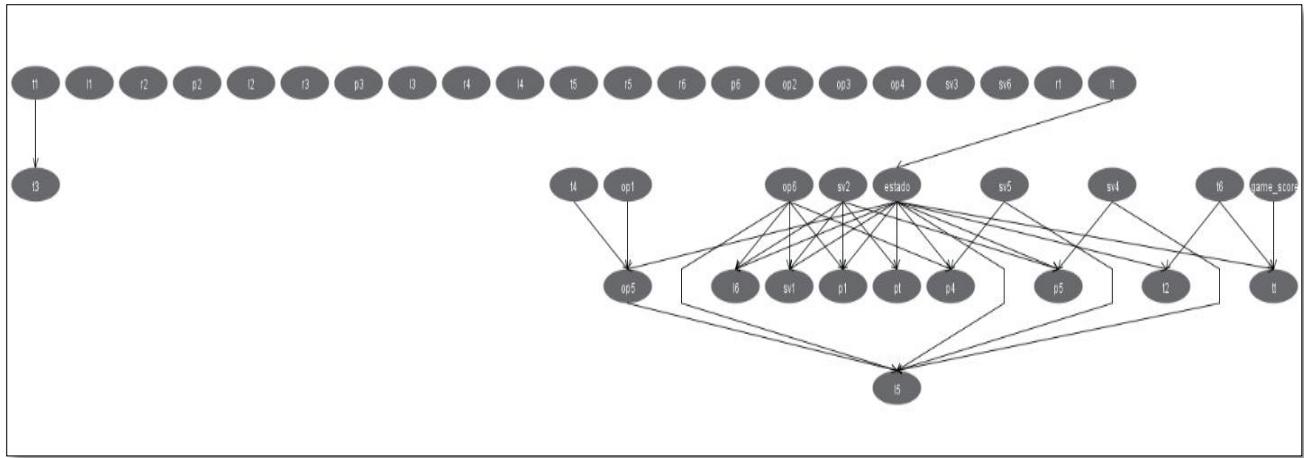


Figura 27: Grafo generado con el algoritmo seleccionado Hill Climber Global balanceado.

Respecto al segundo experimento, el cual considera las 24 variables de los experimentos de Python y R, y considerando los antecedentes de rendimiento de los algoritmos en el experimento anterior, se genera el modelo con el algoritmo Hill Climber Global usando el estimador MLE y el estimador Simple con alfa 0.5, resultando con este último mejores métricas, como puede verse en la tabla 42.

Estimador	Tasa VP	Tasa FP	Precisión	Recall	F-Measure	MCC	Área ROC	Área PRC
Alfa 0 (MLE)	0,891	0,691	0,868	0,891	0,876	0,267	0,594	0,842
Alfa 0.5	0,895	0,643	0,876	0,895	0,882	0,326	0,751	0,892

Tabla 42: Comparación de métricas de algoritmo Hill Climber Global para dos estimadores de probabilidades.

Por lo tanto, se selecciona el modelo con estimador simple para balancear. La matriz de confusión muestra sólo 14 instancias de la clase “Reprobado” correctamente clasificadas, entonces al balancear se espera que este número aumente. Después de probar varios valores para la penalización, se encuentra que con el valor 4 llegan a clasificarse correctamente 25 instancias de la clase “Reprobado”, y algunas métricas presentan mejora. Esto puede verse en la tabla 43.

Modelo	Tasa VP	Tasa FP	Precisión	Recall	F-Measure	MCC	Área ROC	Área PRC
Desbalanceado	0,895	0,643	0,876	0,895	0,882	0,326	0,751	0,892
Balanceado	0,876	0,447	0,887	0,876	0,881	0,396	0,745	0,897

Tabla 43: Comparación de métricas de modelo seleccionado desbalanceado y balanceado.

En la figura 28 es mostrado el grafo generado a partir del modelo balanceado, donde puede observarse la existencia de arcos y, por lo tanto, las posibles relaciones directas entre la clase “estado” y otras variables. Los nodos que representan los resultados de la prueba diagnóstico aparecen aislados, por lo tanto, no representan algún posible tipo de influencia sobre la clase. A diferencia del experimento anterior, en este modelo sí se muestra una posible relación entre la variable “game_score” y la clase “estado”.

También puede observarse que este grafo coincide con los experimentos en *Python* y *R*, en cuanto a la presentación de relaciones directas entre la clase y las variables “programa” y “sol1”, lo que confirma la fuerte influencia que tienen la carrera y la nota de la primera solemne del curso de programación sobre el estado de reprobación / aprobación del estudiante.

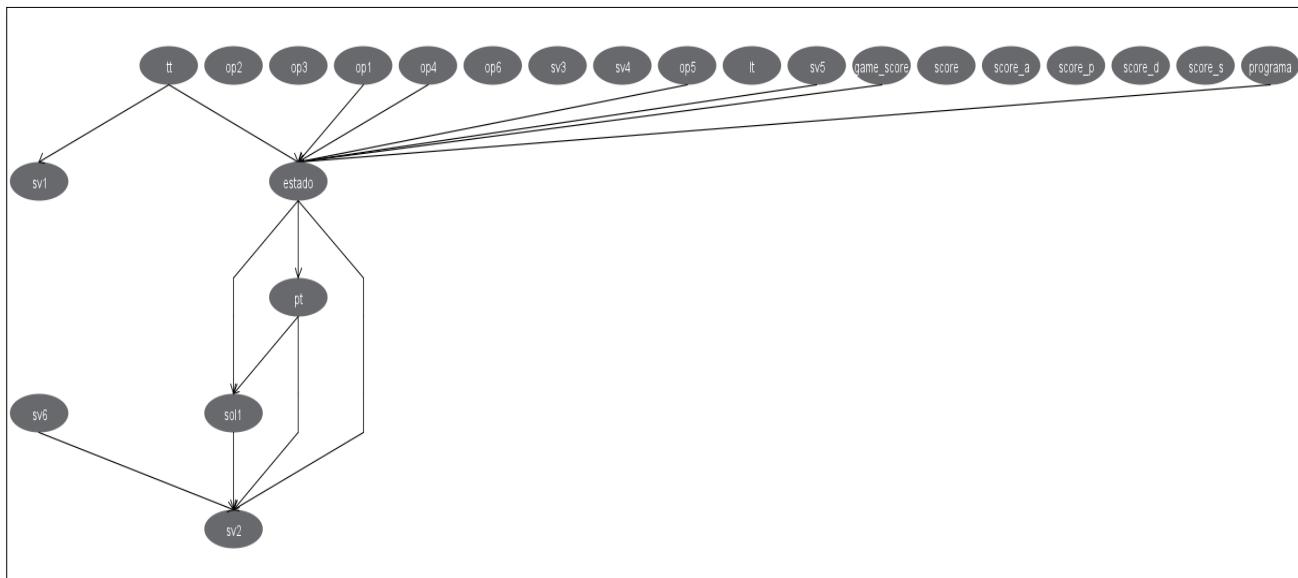


Figura 28: Grafo generado con el modelo seleccionado balanceado.

6.5 Requerimientos Técnicos

La construcción de los modelos Bayesianos tuvo una fuerte componente de programación en lenguaje Python y R. En este apartado del documento, se expondrán todos los requerimientos técnicos que fueron necesarios en el desarrollo de los modelos, desde los requisitos mínimos de hardware, la configuración del entorno de desarrollo para una correcta ejecución y las librerías utilizadas.

6.5.1 Requerimientos para la ejecución de los modelos

Para lograr una adecuada ejecución de los modelos Bayesianos se recomienda los unos requerimientos de hardware y software en particular. Adicionalmente, se explica cómo debe ser configurado el entorno para la ejecución de los experimentos, así como también, las librerías necesarias.

6.5.1.1 Requerimientos de Hardware

- CPU: mas de 8 núcleos
- RAM: 16 GB (se recomiendan 32 GB)
- HDD: 5 GB para instalación de software necesario (se recomienda espacio adicional para los experimentos)
- Acceso a Internet para la descarga de las librerías

6.5.1.2 Requerimientos de Software

- El entorno cliente puede ser una arquitectura de sistema Windows 64 bits, macOS 64 bits o Linux
- Anaconda Navigator: La instalación provee de Python y R creando entornos para el desarrollo de software
- WEKA (Waikato Environment for Knowledge Analysis), para pruebas rápidas sin requerir programación.

6.5.2 Configuración de ambiente

La configuración del entorno de trabajo está ligado principalmente a la instalación del software Anaconda Edición Individual. Este software es posible descargarlo y seguir los

pasos de instalación desde su sitio web² para el caso de un sistema Windows, que fue lo utilizado este proyecto.

Paso siguiente, desde Anaconda Navigator se debe crear un entorno nuevo, se recomienda utilizar un nombre descriptivo como, por ejemplo, “BNLEARN”.

Paso final, se deben instalar las librerías que fueron utilizadas para el desarrollo del programa Python.

Para la programación de los modelos Bayesianos se recomienda utilizar Jupyter Notebooks para el análisis de datos 1D / 2D y para el programa principal se puede utilizar Jupyter Lab que provee un entorno más integrado. También, es recomendable utilizar Visual Studio Code desde Anaconda Navigator.

JupyterLab es un 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)

6.5.3 Librerías utilizadas

Se utilizaron 11 librerías que serán expuestas aquí en orden de importancia:

- **Bnlearn:** Paquete Python para aprender la estructura gráfica de redes Bayesianas, aprendizaje de parámetros, inferencia y métodos de muestreo. (Taskesen, 2019)
- **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)
- **Imblearn:** Es un paquete de Python que ofrece una serie de técnicas de remuestreo utilizadas en conjuntos de datos que poseen un marcado desequilibrio de clases. (Guillaume, Nogueira, & Aridas, 2017)
- **Rpy2:** Interfaz de Python al lenguaje R. rpy2 está ejecutando un R incrustado, proporcionando acceso a él desde Python utilizando la propia C-API de R a través de una interfaz de alto nivel que hace funciones de R y objetos al igual que las

² Sitio web oficial de Anaconda: <https://docs.anaconda.com/anaconda/install/windows/>

funciones de Python y proporciona una conversión perfecta a estructuras de datos numpy y pandas. También proporciona funciones para cuando se trabaja con notebooks Jupyter o Ipython. (Rpy2, s.f.)

- **Graphviz:** Este paquete facilita la creación y representación de descripciones de gráficos en el lenguaje DOT del software de dibujo de gráficos Graphviz de Python. Crea un objeto gráfico, ensambla el gráfico agregando nodos y bordes, y recupera su cadena de código fuente DOT. Guarda el código fuente en un archivo y lo renderiza. (Graphviz, s.f.)
- **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)
- **Astropy:** Es un paquete que contiene funciones clave y herramientas comunes necesarias para realizar astronomía y astrofísica con Python. Es el núcleo del Proyecto Astropy, que tiene como objetivo permitir que la comunidad desarrolle un ecosistema sólido de paquetes afiliados que cubran una amplia gama de necesidades de investigación astronómica, procesamiento de datos y análisis de datos. ({Astropy Collaboration} and {Robitaille}, 2018)
- **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 conocidas 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)

7 CONCLUSIONES

El objetivo fundamental de la tesis era 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 estudió una librería para Python y R que permiten el modelamiento de redes Bayesianas (bnlearn) del cual se determinó que fue su versión en R es la más robusta debido a la amplia gama de parametrizaciones que pueden realizar, además de una mejor optimización a la hora de ejecutar experimentos. Adicionalmente, se utilizó como herramienta software empaquetado, la llamada Weka, que permitió construir un modelo bayesiano y realizar una comparación de los resultados con los experimentos realizados en Python.

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 un modelo con las 10 variables mejor correlacionadas, además de algunas variables que fueron incluidas por recomendación de experto.

En cuanto al quinto objetivo, generar métricas para evaluar la efectividad y desempeño del modelo creado, se puede concluir que se logró ampliamente la obtención de métricas que permitieron identificar dos redes Bayesianas con un alto porcentaje de ajuste, por sobre el 70% en uno y por sobre el 85% en otro.

Sobre el sexto objetivo planteado, evaluar los resultados obtenidos para identificar las causas que desencadenan la reprobación, se puede concluir que los resultados obtenidos permiten determinar que el modelo construido es efectivo y permite identificar las variables que son relevantes para determinar la situación final del curso.

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

En una primera etapa los resultados de las redes bayesianas encontradas determinaron 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, se aplicaron técnicas de balanceo al modelo y se incluyeron métricas puntuales que permitieron mejorar la precisión del modelo.

Las métricas incluidas permitieron medir la sensibilidad y la especificidad como lo es la curva ROC, también la de precisión (precision) que es la fracción de instancias relevantes recuperadas y la de recuperación (recall) llamada sensibilidad. Con estas métricas se pudo tener una visión mejorada de cómo se comportaba el modelo permitiendo encontrar aquellas redes que mejor comportamiento tenían a medida que se aplicaban técnicas como las de selección de variables y restricciones de arcos.

Con lo anterior resuelto se continuó con la estratificación de los datos en 5 particiones, 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 en el caso de hacer uso de la base de datos desbalanceada, pero si se aplican las técnicas de balanceo la porción de entrenamiento aumenta al doble los registros a utilizar. En términos de resultado, los valores de exactitud para los mejores modelos encontrados nos indican un 74,5% de confianza para un modelo que usa datos desbalanceados y con una selección de 5 variables y para el segundo modelo escogido el porcentaje de exactitud aumentó a un 88,9% usando datos balanceados y con restricciones de arcos.

En concreto, durante los primeros experimentos, utilizando la base de datos desbalanceada, el modelo mostraba un alto porcentaje de confianza, pero al contener ese sesgo, con el potente desequilibrio de la clase, llevó a la implementación de técnicas que permitieran corroborar las primeras evidencias. Estas técnicas, de balanceo de clases, selección de variables y aplicación de restricciones, confirmaron la primera evidencia y, es más, permitieron aumentar el nivel de confianza del modelo subiendo los porcentajes de exactitud.

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 preprocesar datos, configurar un experimento, ejecutar el experimento y obtener resultados, evitando las dificultades y obstáculos que pueden surgir de la programación. Para el tamaño de la muestra de datos objetivo de estudio, resultó ser una herramienta rápida y eficiente, aunque también esto depende del algoritmo utilizado, ya que en el experimento de 41 variables con Repeated Hill Climber usando como parámetro 10 iteraciones, la ejecución toma varias horas debido al procesamiento requerido.

Weka nos entrega varias medidas de rendimiento del modelo, entre ellas está la TP Rate, FP Rate, Precisión, Recall, F-Measure, MCC, Área ROC y Área PRC. 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 0,660 no muy satisfactorio para el primer experimento y 0,876 para el segundo, que es considerado 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 0,628 en el primer experimento y 0,447 en el segundo, indicando una probabilidad cercana a la 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 0,816 en el primer experimento y 0,887 en el segundo, es bastante alta considerando que de 8 de 10 casos serían 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 0,660 en el primer experimento, lo cual no es muy satisfactorio, y 0,876 en el segundo, lo que indica que la mayoría de 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 0,721 en el primer experimento y 0,881 en el segundo, indicando una puntuación bastante alta.
- Área 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 0,507 en el primer experimento y 0,745 en el segundo, siendo por tanto este último de un mejor desempeño que el anterior.

El modelo del primer experimento, al incluir solamente las variables del juego diagnóstico, y a pesar de no mostrar dentro de sus métricas un Área ROC satisfactoria, permite ver relaciones con el estado de reprobación / aprobación del estudiante, por lo tanto, se puede concluir que los resultados del juego diagnóstico sí pueden predecir sobre la reprobación del curso de programación por parte del estudiante.

El modelo analizado en el segundo experimento, al considerar las mismas variables que los experimentos en Python y R, coincide con ellos respecto a las relaciones encontradas. Se confirma la estrecha relación entre la carrera del estudiante y el hecho de que el estudiante apruebe o repreuebe el curso de programación. En cuanto a la prueba diagnóstico, no se puede concluir que sus resultados puedan predecir el desempeño del estudiante en el curso de programación mediante este modelo, pero del juego diagnóstico sí se puede afirmar que existe la posibilidad de que sus métricas puedan deducir el futuro desempeño del estudiante en el curso de programación.

8 REFERENCIAS BIBLIOGRÁFICAS

- {Astropy Collaboration} and {Robitaille}, T. a.-W. (09 de 2018). The Astropy Project: Building an Open-science Project and Status of the v2.0 Core Package. 156(3), 123.
- 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>
- Akaike, H. (1998). Information Theory and an Extension of the Maximum Likelihood Principle. En E. Parzen, & K. Tanabe (Edits.), *Selected Papers of Hirotugu Akaike* (págs. 199-213). New York, NY. doi:10.1007/978-1-4612-1694-0_15
- Ankan, A. a. (2015). Probabilistic graphical models using python. En Citeseer (Ed.), *Proceedings of the 14th Python in Science Conference (SCIPY 2015)*. Obtenido de <https://github.com/pgmpy/pgmpy>
- 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>
- Chawla, N., Bowyer, K., Hall, L., & Kegelmeyer, W. (2002). SMOTE: Synthetic Minority Over-sampling Technique. *Journal of Artificial Intelligence Research*, 321-357.

Cooper, G., & Herskovits, E. (1992). A Bayesian Method for the Induction of Probabilistic Networks from Data. *Machine Learning*, 9, 309-347.

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., & Smyth, P. (1996). The KDD Process for Extracting Useful Knowledge from Volumes of Data. *Communications of the ACM*, 39(11), 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.

Graphviz. (s.f.). Graphviz. Obtenido de <https://graphviz.readthedocs.io/en/stable/index.html>

Guillaume, L., Nogueira, F., & Aridas, C. K. (2017). Imbalanced-learn: A Python Toolbox to Tackle the Curse of Imbalanced Datasets in Machine Learning. *Journal of Machine Learning Research*, 18(17), 1-5. Obtenido de <http://jmlr.org/papers/v18/16-365>

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.

Heckerman, D., Geiger, D., & Chickering, D. (1995). Learning Bayesian Networks: The Combination of Knowledge and Statistical Data. *Machine Learning*, 20, 197 - 243.

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

Kass, R., & Raftery, A. (06 de 1995). Bayes Factors. *Journal of the American Statistical Association*, 90(430), 773-795. Obtenido de <http://links.jstor.org/sici?&sici=0162-1459%28199506%2990%3A430%3C773%3ABF%3E2.0.CO%3B2-8>

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.

Ludeman, R. B., & Schreiber, B. (2020). *Student Affairs and Services in Higher Education: Global Foundations, Issues, and Best Practices* (Tercera ed., Vol. VIII). Berlin, Alemania: International Association of Student Affairs and Services (IASAS) - Deutsches Studentenwerk (DSW).

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.

Mesa Páez, L., Rivera Lozano, M., & Romero Davila, J. (02 de 2011). Descripción general de la Inferencia Bayesiana y sus aplicaciones en los procesos de gestión. (U. d. Rosario, Ed.) *La Simulación al Servicio de la Academia*, 2, 1-28. Obtenido de https://www.urosario.edu.co/urosario_files/38/38e60ea0-497e-4197-913d-e156ae0bb084.pdf

Minerva. (s.f.). *Minerva*. Recuperado el 2021, de <https://mnrv.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. *Revista Latinoamericana de Ingeniería del Software*, 2(5). doi:<https://doi.org/10.18294/relais.2014.273>-306
- 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>
- Ramírez Hereza, P., & Ramos Castro, D. (2020). *Redes Bayesianas para predicción y descubrimiento de relaciones con señales procedentes de sensores industriales*. Universidad Autónoma de Madrid, Escuela Politécnica Superior.
- 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=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>
- Rpy2, L. G. (s.f.). *rpy2 - R en Python*. Obtenido de <https://rpy2.github.io/index.html>
- 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 para la predicción del rendimiento académico mediante técnica de clasificación bayesiana. *Revista Dilemas Contemporáneos: Educación, Política y Valores*(3), 1-23.
- SAS Institute. (1998). Data Mining and the Case for Sampling.

- 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.
- Schwarz, G. (03 de 1978). Estimating the Dimension of a Model. *The Annals of Statistics*, 6, 461-464.
- Scutari, M. (2010). Learning Bayesian Networks with the bnlearn R Package. Obtenido de <https://arxiv.org/abs/0908.3817>; <https://www.bnlearn.com/>
- SIES. (2014). *Panorama de la Educación Superior en Chile*. División de Educación Superior, Ministerio de Educación.
- SIES. (2020). *Informe Retención de 1er año de pregrado*. División de Educación Superior, Ministerio de Educación. Obtenido de <https://www.mifuturo.cl/informes-retencion-de-primer-ano/>
- 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>
- team, T. p. (02 de 2020). *Pandas*. Obtenido de <https://pandas.pydata.org/>
- UNAB. (2019). *Self-Study Report prepared for the Middle States Commission on Higher Education (MSCHE)*. Universidad Andrés Bello.
- UNESCO-IESALC. (2020). *Towards universal access to higher education: international trends*. Instituto Internacional para la Educación Superior en América Latina y el Caribe (UNESCO-IESALC).
- 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.

9 ANEXOS

9.1 DAGs de los experimentos ejecutados con R sobre Python

Se presenta una muestra de los DAGs, de la primera partición, obtenidos de todos los experimentos ejecutados con R sobre Python.

9.1.1 Experimento con datos balanceados y discretos con medida de puntuación AIC

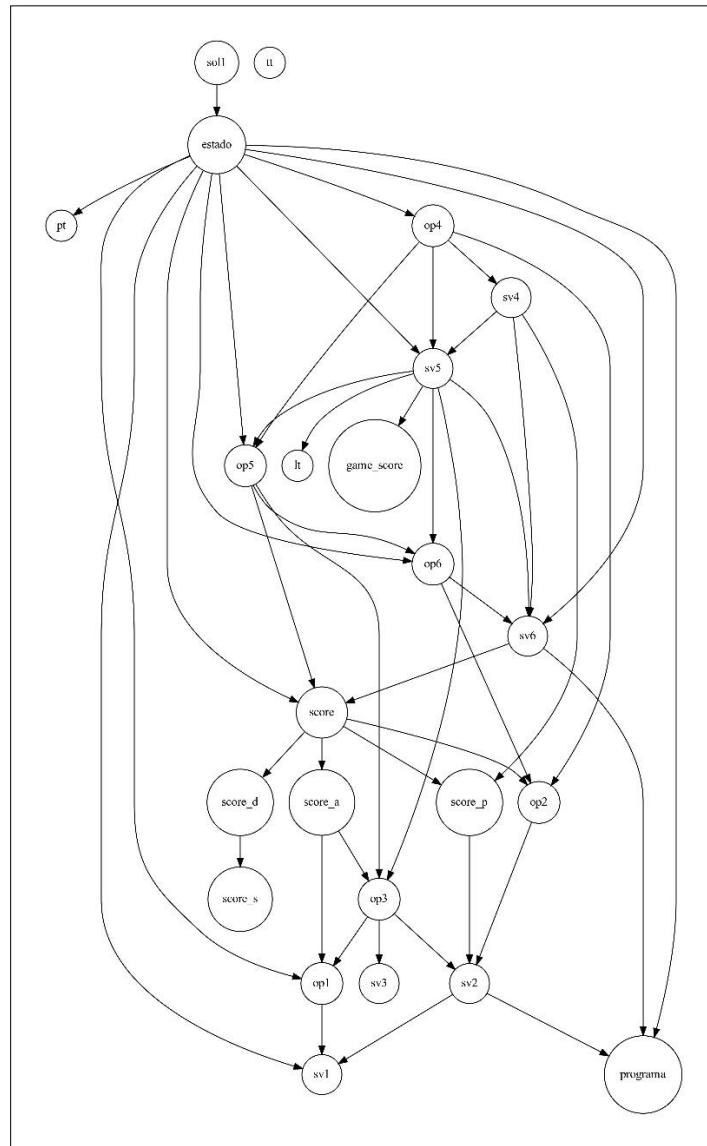


Figura 29: Red de la partición 1 del experimento con datos balanceados y discretos con medida de puntuación AIC.

9.1.2 Experimento con datos balanceados y discretos con medida de puntuación

BIC

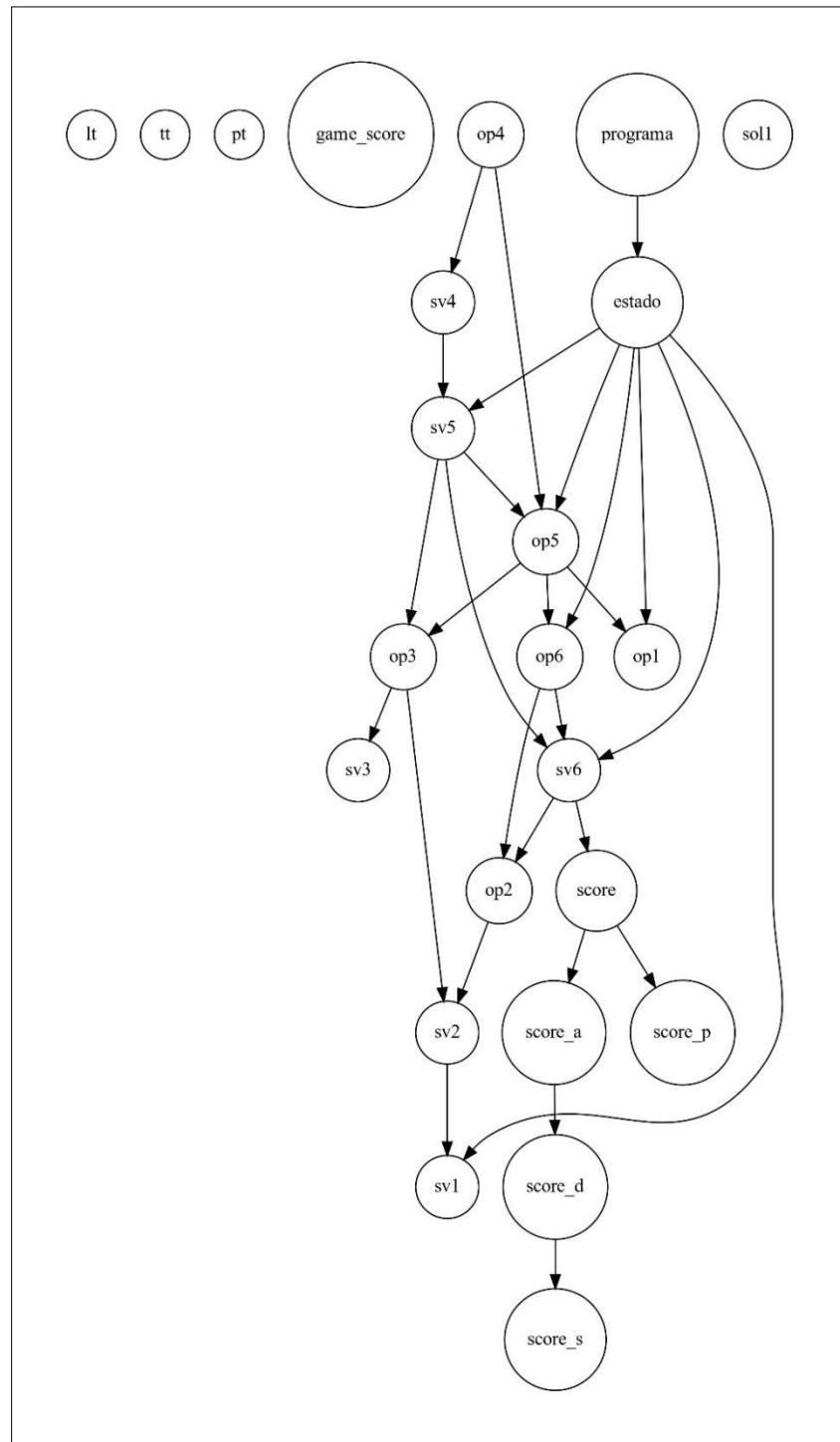


Figura 30: Red de la partición 1 del experimento con datos balanceados y discretos con medida de puntuación BIC.

9.1.3 Experimento con datos balanceados, discretos y continuos con medida de puntuación AIC-CG

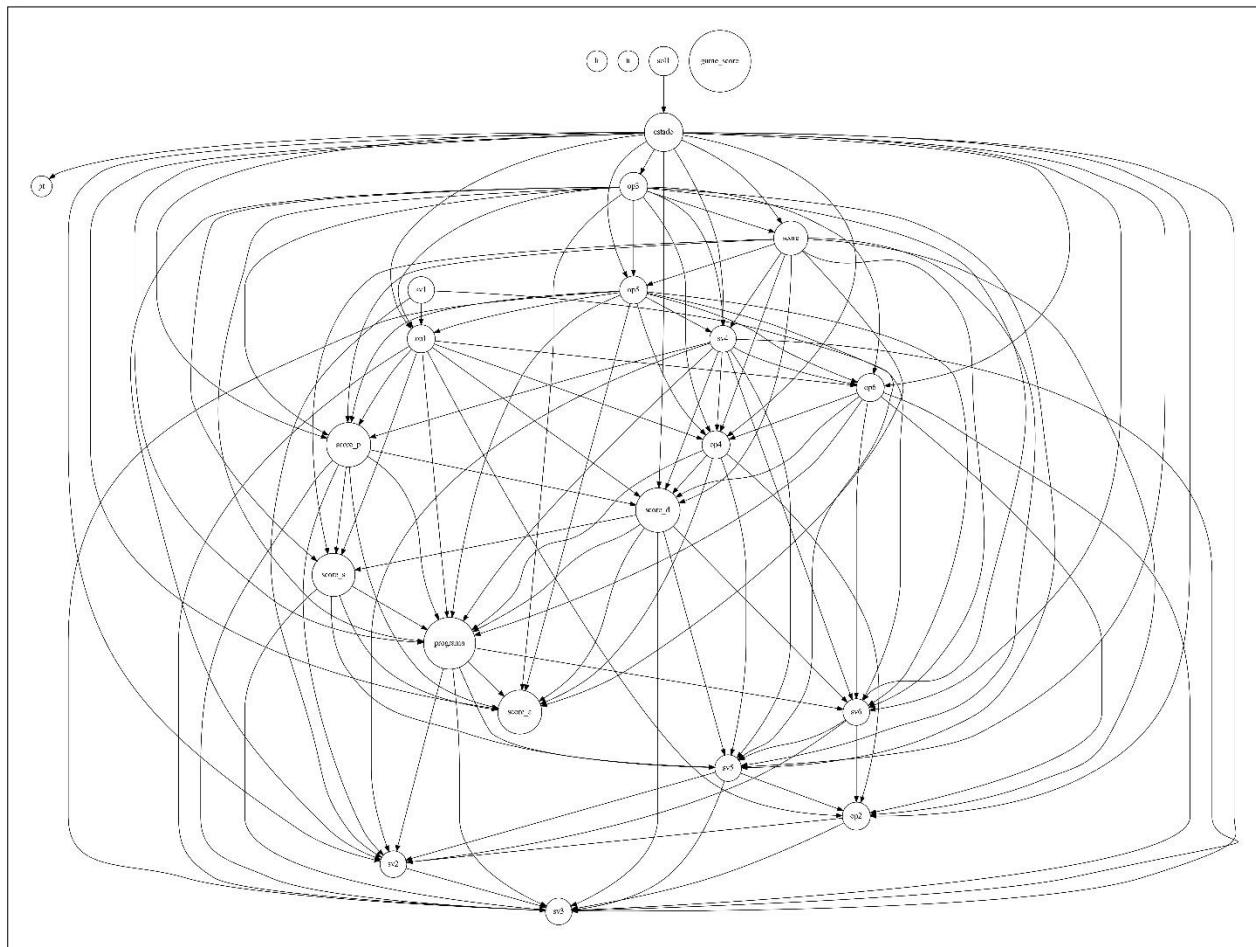


Figura 31: Red de la partición 1 del experimento con datos balanceados, discretos y continuos con medida de puntuación AIC-CG.

9.1.4 Experimento con datos balanceados, discretos y continuos con medida de puntuación AIC-CG con selección de 5 variables

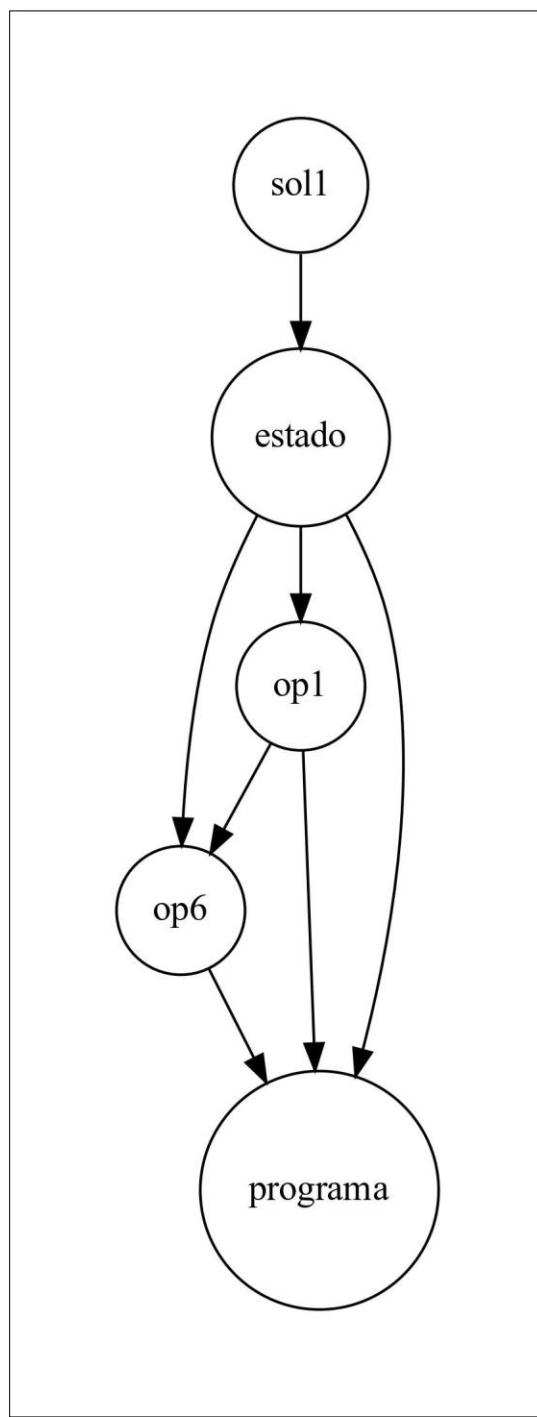


Figura 32: Red de la partición 1 del experimento con datos balanceados, discretos y continuos con medida de puntuación AIC-CG con selección de 5 variables.

9.1.5 Experimento con datos balanceados, discretos y continuos con medida de puntuación AIC-CG con selección de 10 variables

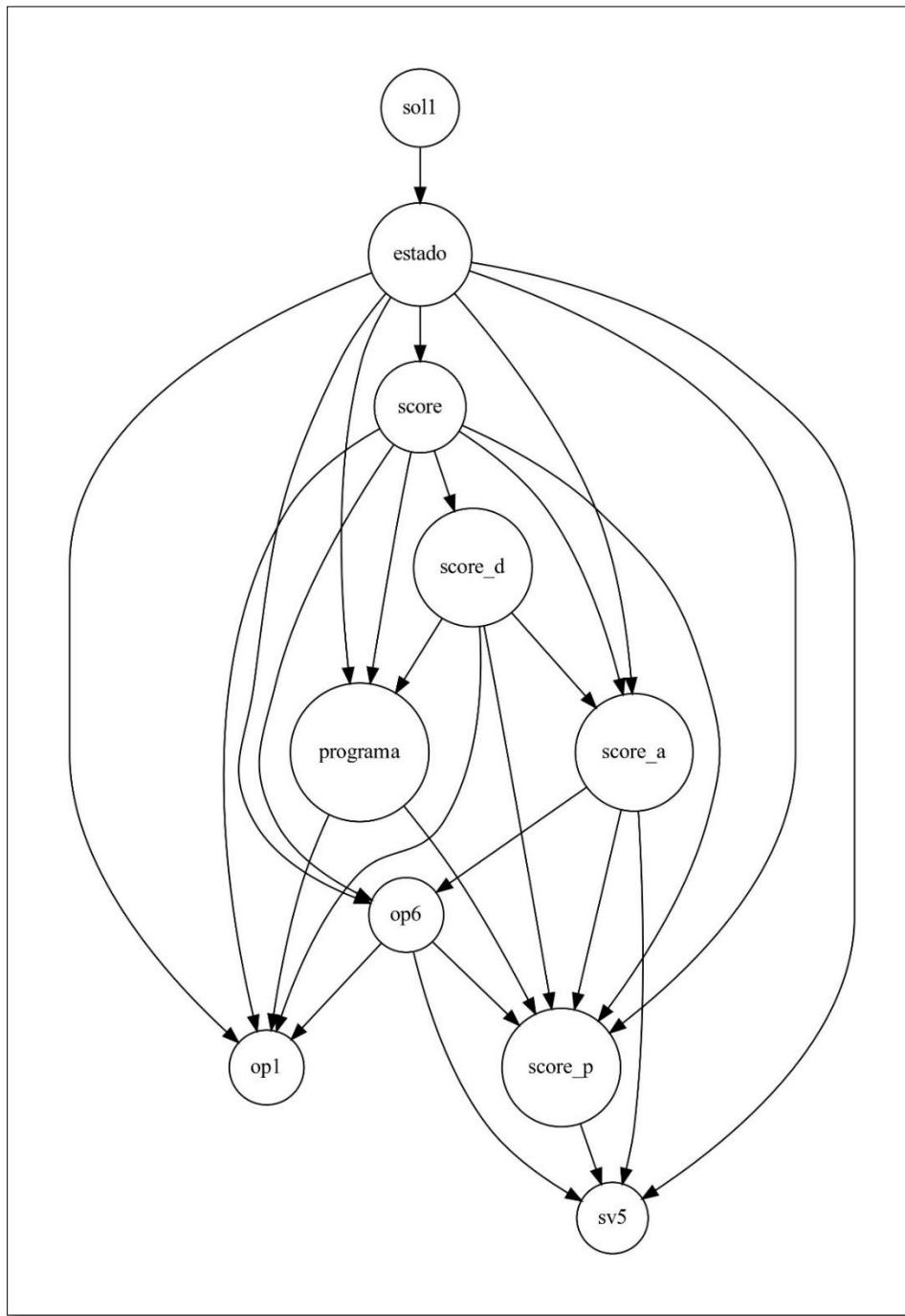


Figura 33: Red de la partición 1 del experimento con datos balanceados, discretos y continuos con medida de puntuación AIC-CG con selección de 10 variables.

9.1.6 Experimento con datos balanceados, discretos y continuos con medida de puntuación AIC-CG con selección de 15 variables

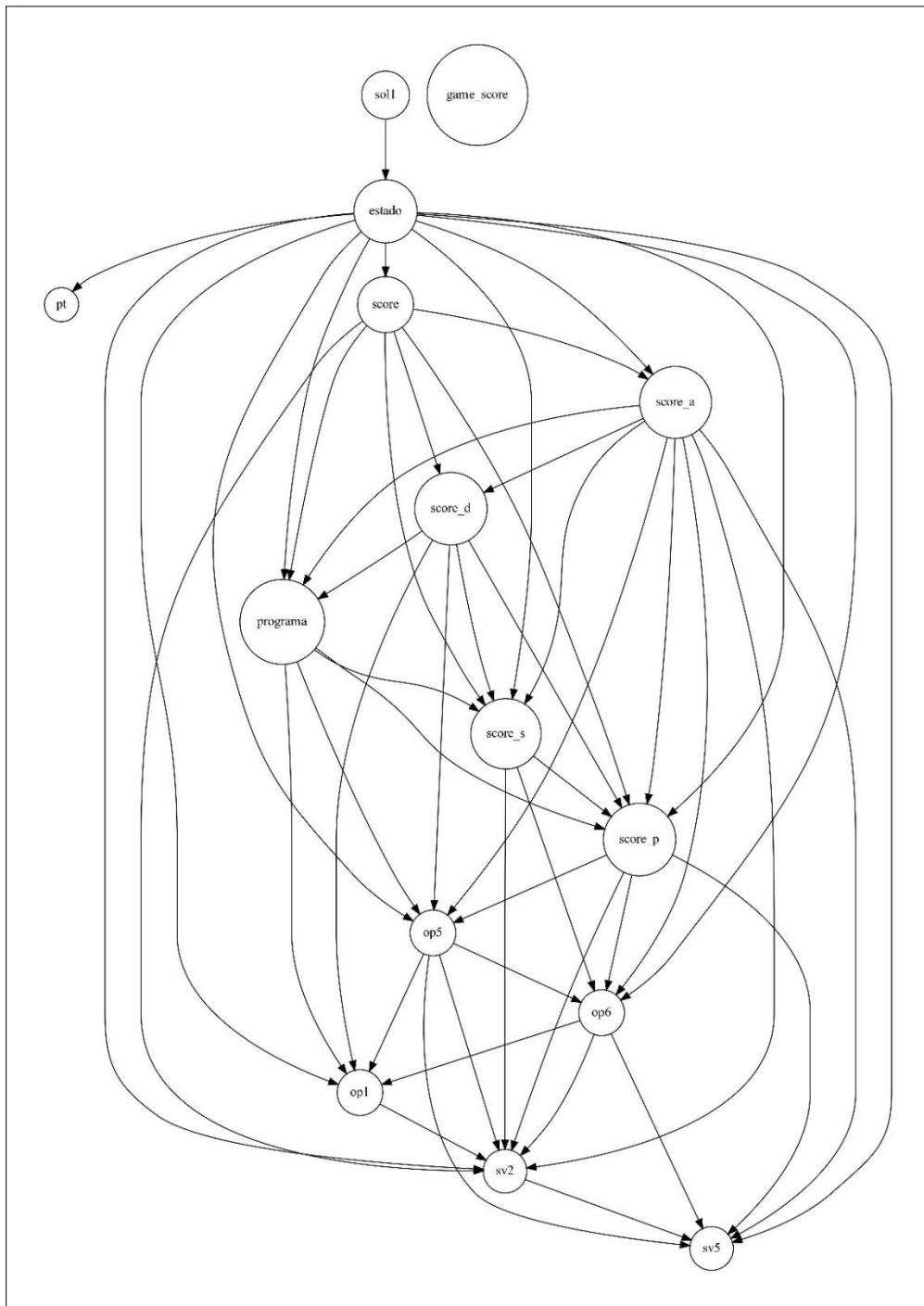


Figura 34: Red de la partición 1 del experimento con datos balanceados, discretos y continuos con medida de puntuación AIC-CG con selección de 15 variables.

9.1.7 Experimento con datos balanceados, discretos y continuos con medida de puntuación AIC-CG con selección de 20 variables

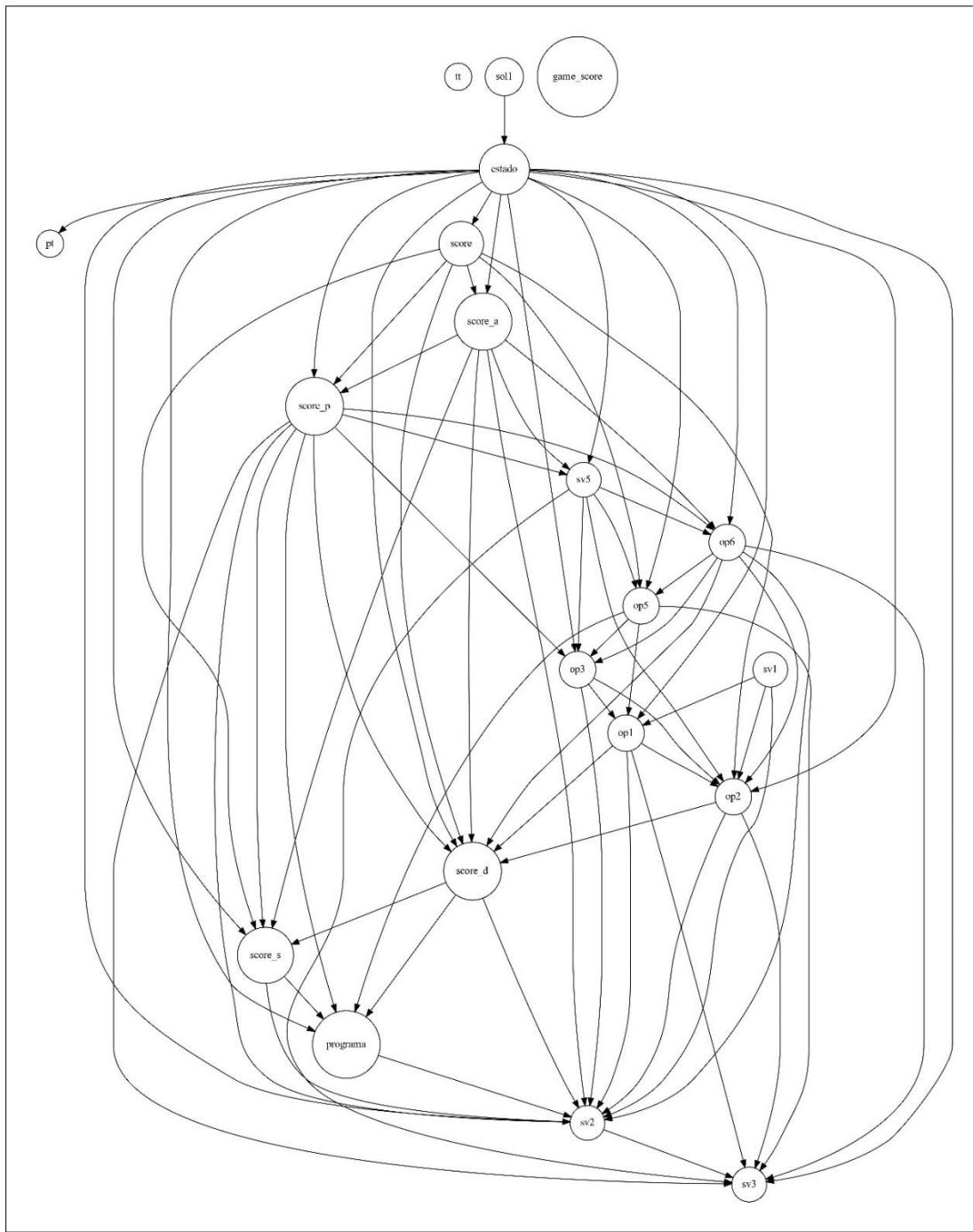


Figura 35: Red de la partición 1 del experimento con datos balanceados, discretos y continuos con medida de puntuación AIC-CG con selección de 20 variables.

9.1.8 Experimento con datos balanceados, discretos y continuos con medida de puntuación AIC-CG con aplicación de restricción de arcos

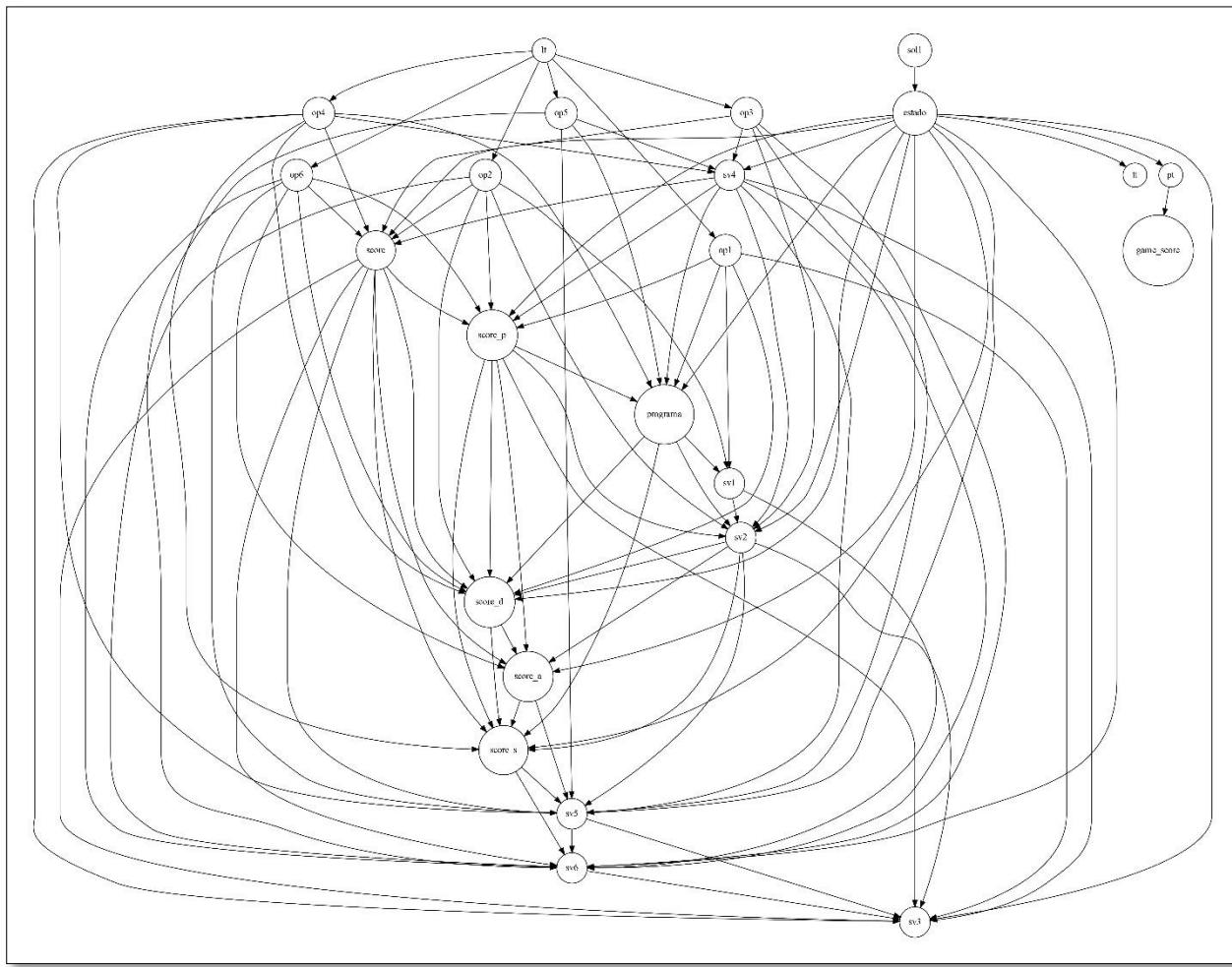


Figura 36: Red de la partición 1 del experimento con datos balanceados, discretos y continuos con medida de puntuación AIC-CG con aplicación de restricción de arcos.

9.1.9 Experimento con datos balanceados, discretos y continuos con medida de puntuación BIC-CG

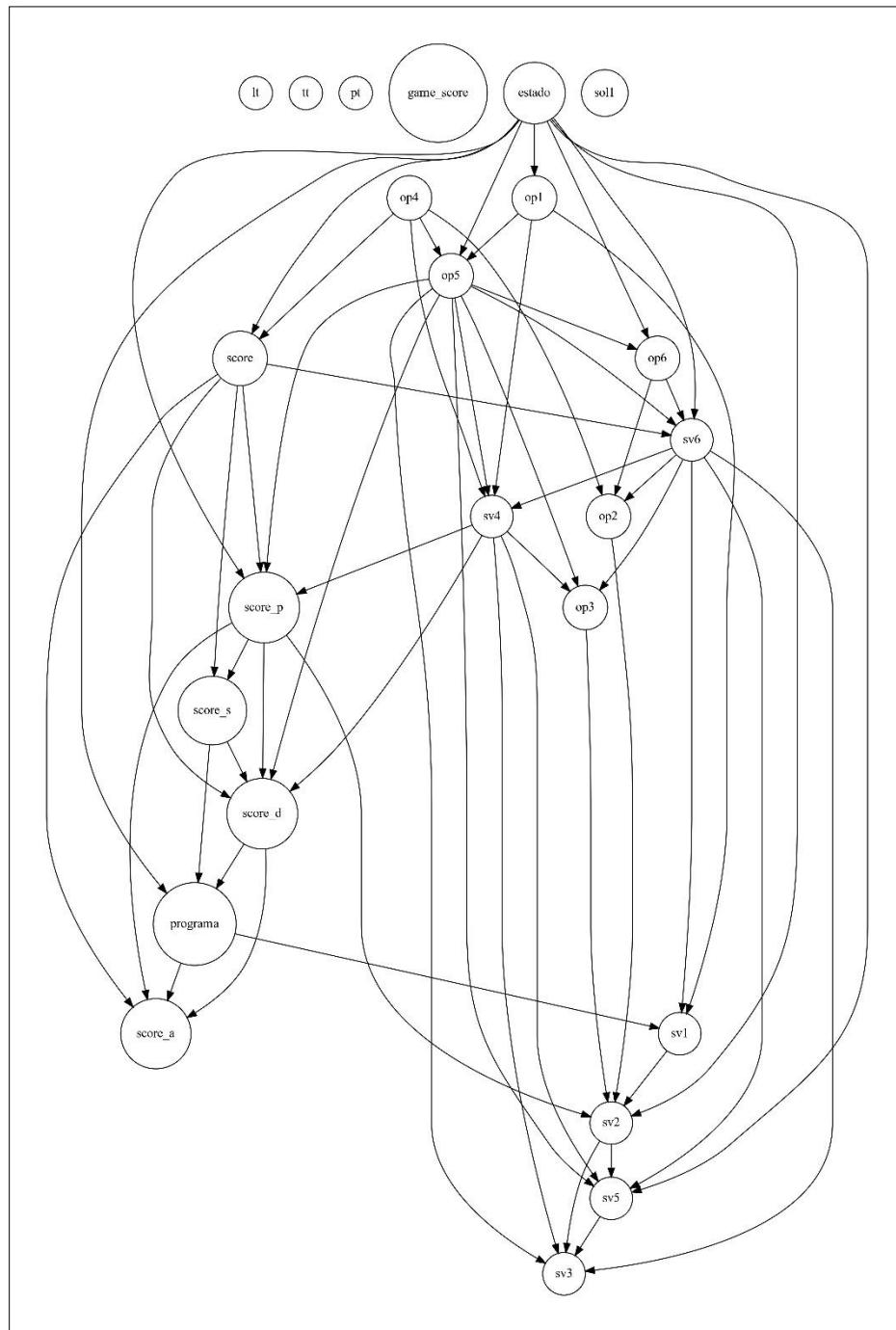


Figura 37: Red de la partición 1 del experimento con datos balanceados, discretos y continuos con medida de puntuación BIC-CG.

9.1.10 Experimento con datos desbalanceados y discretos con medida de puntuación AIC

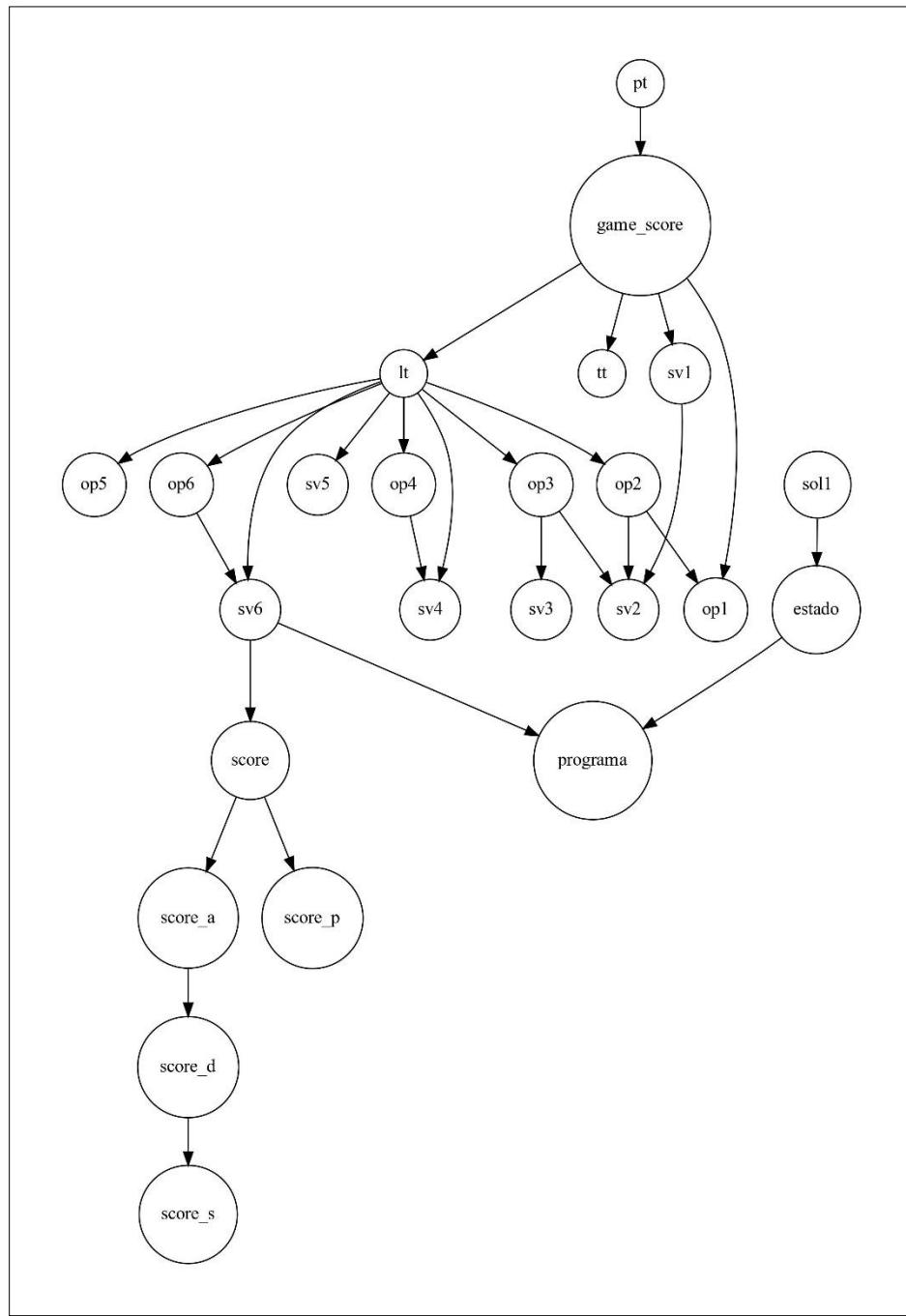


Figura 38: Red de la partición 1 del experimento con datos desbalanceados y discretos con medida de puntuación AIC.

9.1.11 Experimento con datos desbalanceados y discretos con medida de puntuación AIC con selección de 5 variables

De este experimento se tomó la partición 2 dado que están todos sus nodos conectados.

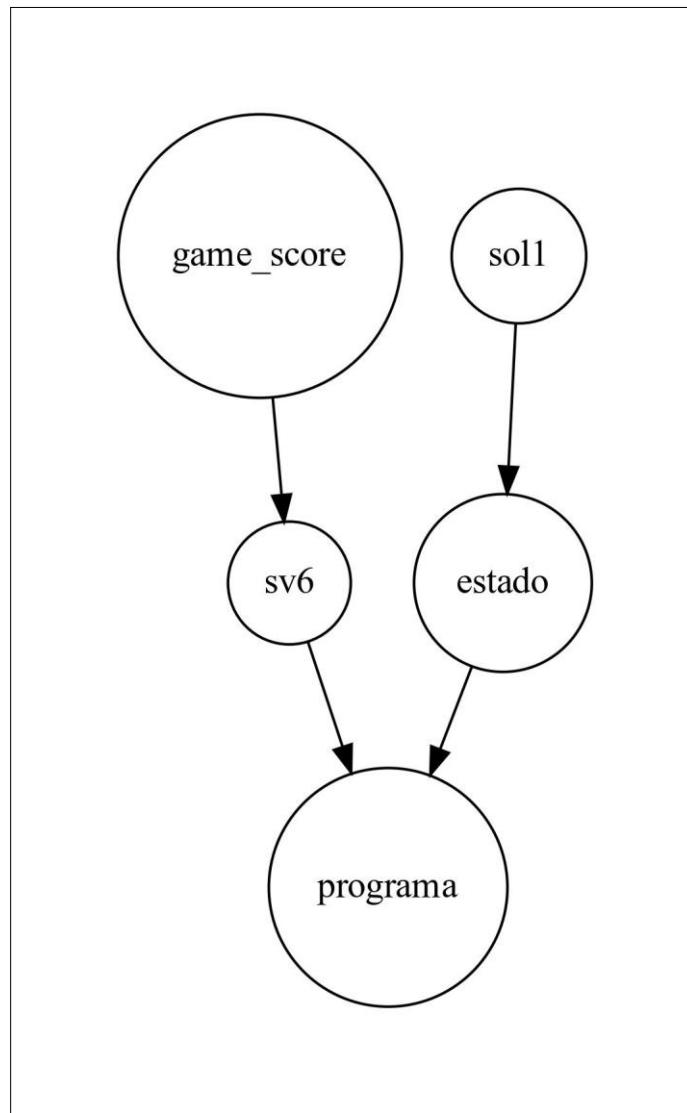


Figura 39: Red de la partición 2 del experimento con datos desbalanceados y discretos con medida de puntuación AIC con selección de 5 variables.

9.1.12 Experimento con datos desbalanceados y discretos con medida de puntuación AIC con selección de 10 variables

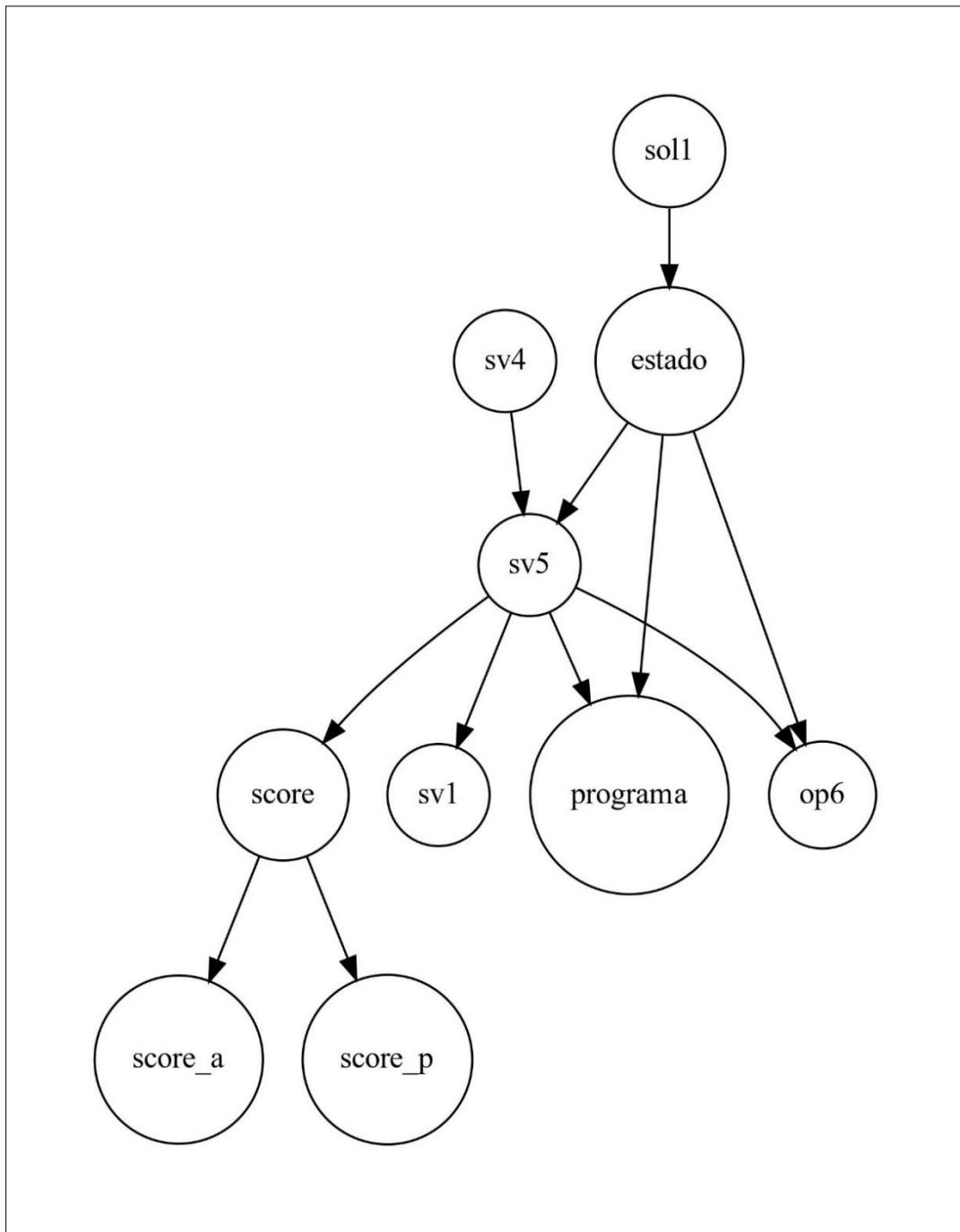


Figura 40: Red de la partición 1 de experimento con datos desbalanceados y discretos con medida de puntuación AIC con selección de 10 variables.

9.1.13 Experimento con datos desbalanceados y discretos con medida de puntuación AIC con selección de 15 variables

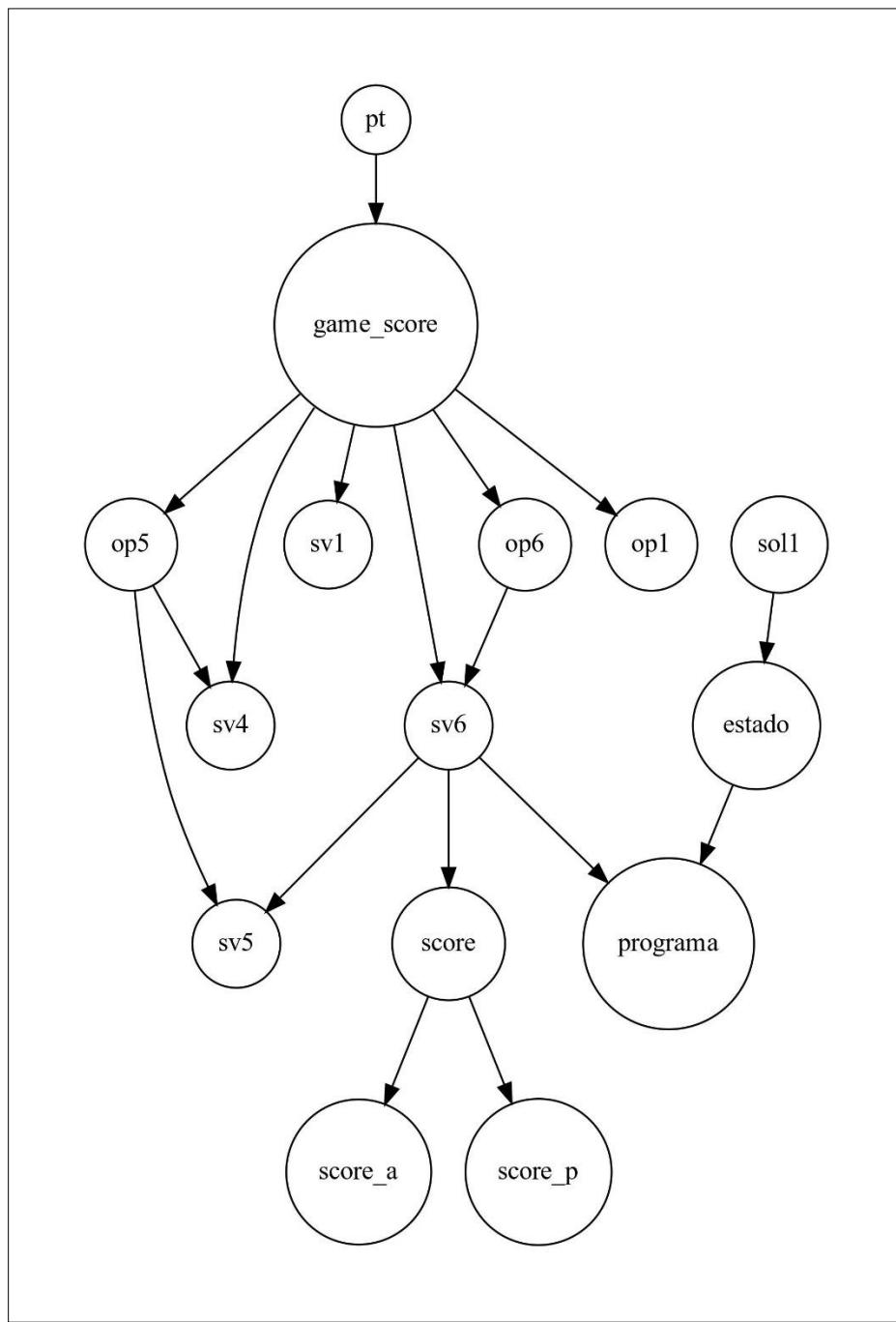


Figura 41: Red de la partición 1 del experimento con datos desbalanceados y discretos con medida de puntuación AIC con selección de 15 variables.

9.1.14 Experimento con datos desbalanceados y discretos con medida de puntuación AIC con selección de 20 variables

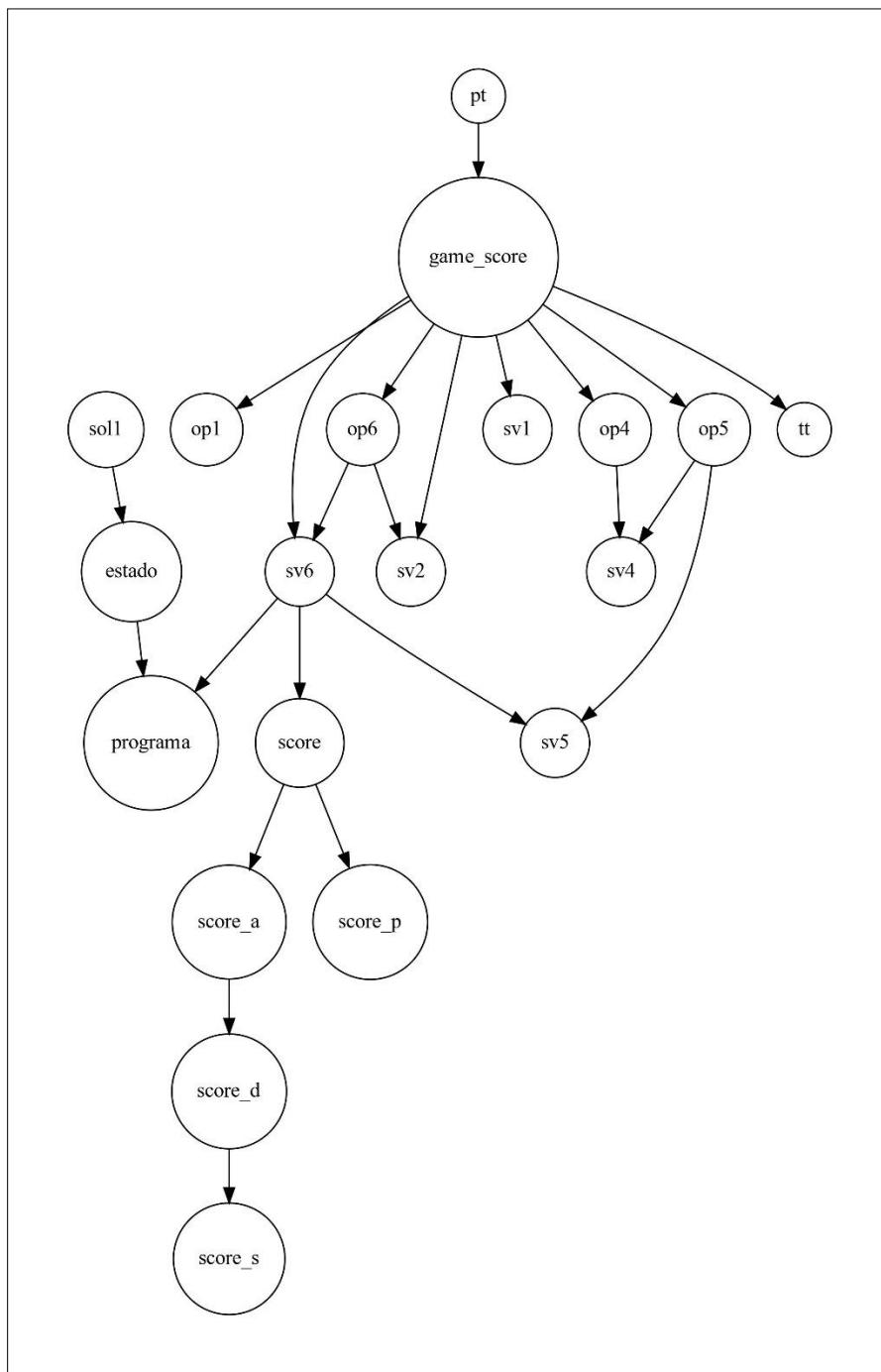


Figura 42: Red de la partición 1 del experimento con datos desbalanceados y discretos con medida de puntuación AIC con selección de 20 variables.

9.1.15 Experimento con datos desbalanceados y discretos con medida de puntuación AIC con aplicación de restricción de arcos

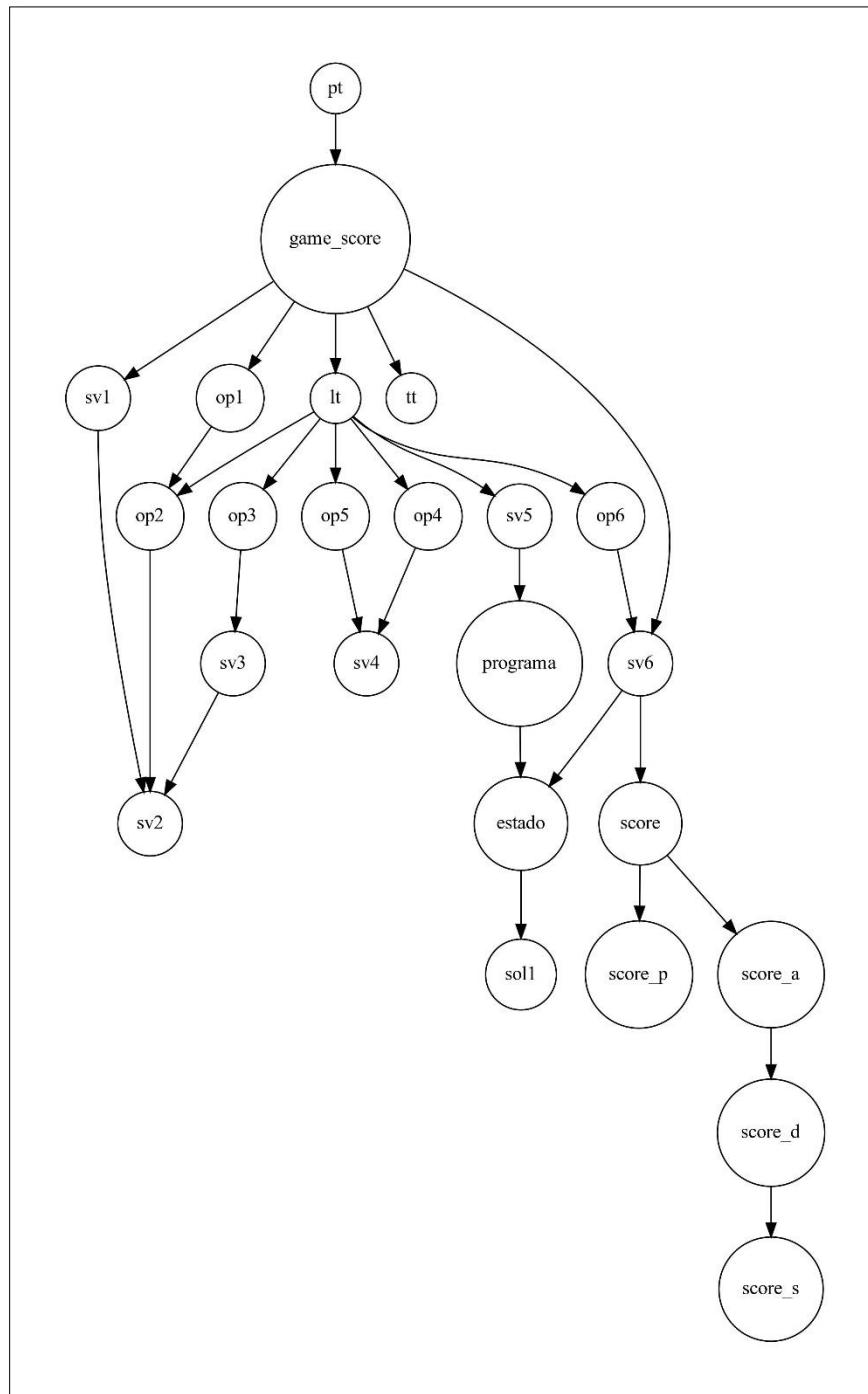


Figura 43: Red de la partición 1 del experimento con datos desbalanceados y discretos con medida de puntuación AIC con aplicación de restricción de arcos.

9.1.16 Experimento con datos desbalanceados y discretos con medida de puntuación BIC

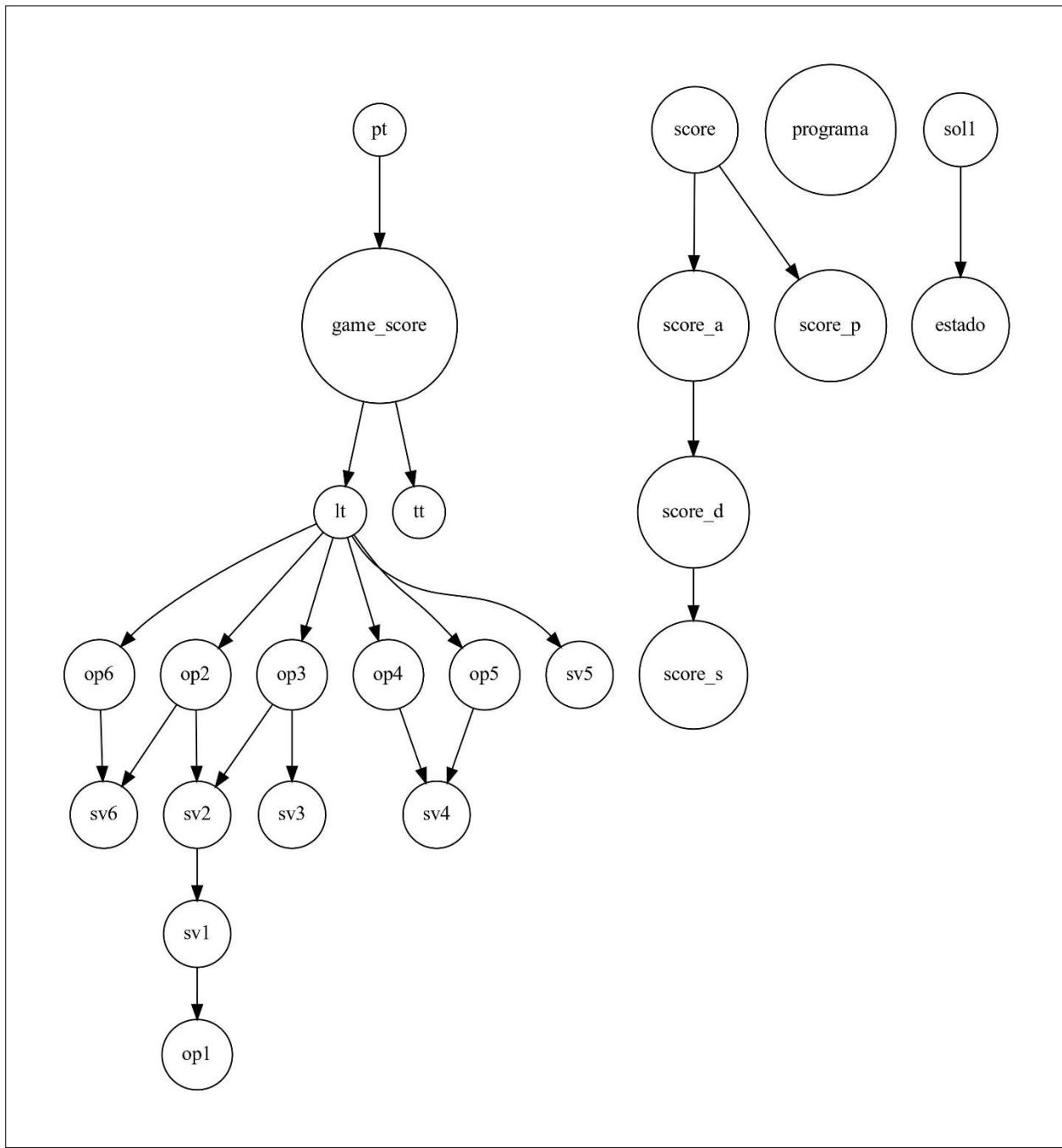


Figura 44: Red de la partición 1 del experimento con datos desbalanceados y discretos con medida de puntuación BIC.

9.1.17 Experimento con datos desbalanceados, discretos y continuos con medida de puntuación AIC-CG

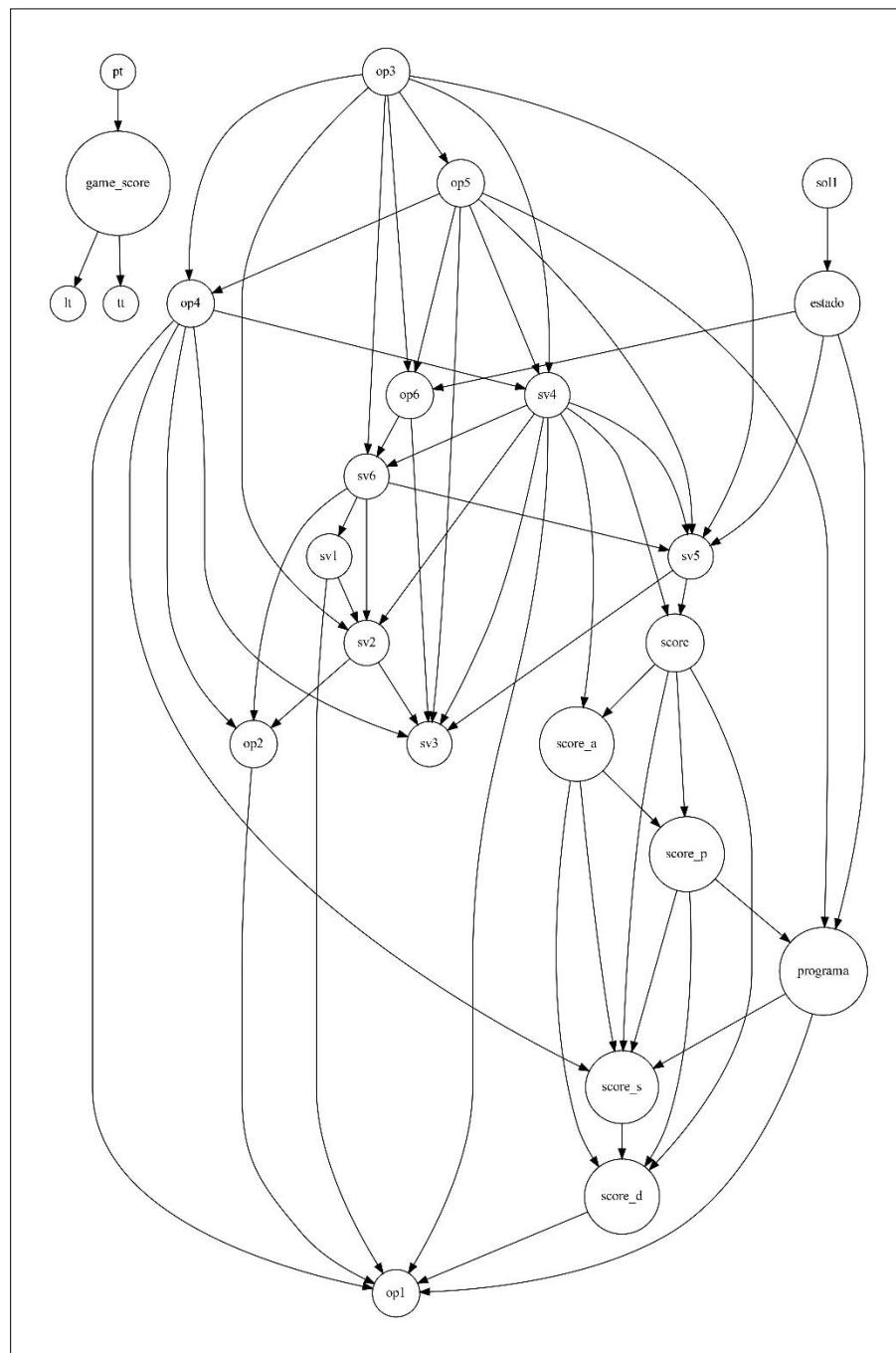


Figura 45: Red de la partición 1 del experimento con datos desbalanceados, discretos y continuos con medida de puntuación AIC-CG.

9.1.18 Experimento con datos desbalanceados, discretos y continuos con medida de puntuación BIC-CG

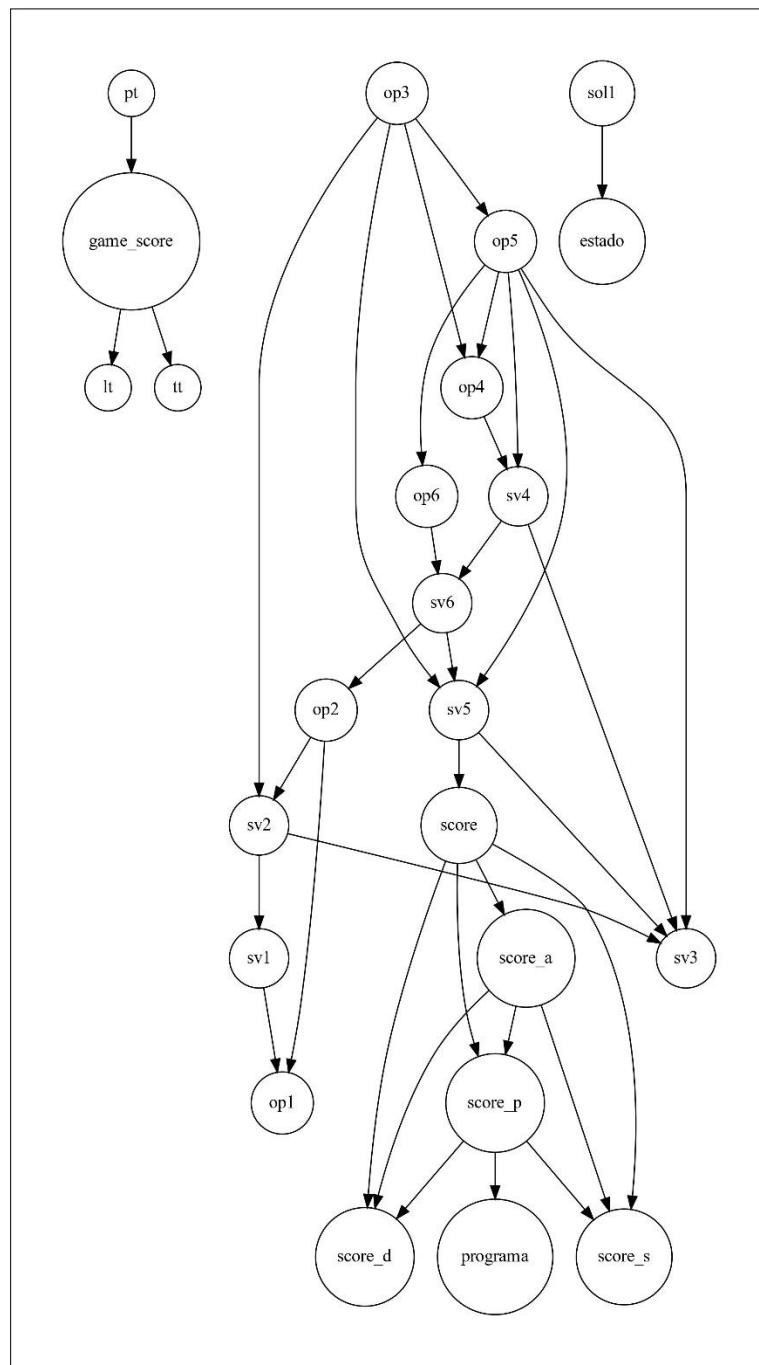


Figura 46: Red de la partición 1 del experimento con datos desbalanceados, discretos y continuos con medida de puntuación BIC-CG.

9.1.19 Experimento con datos desbalanceados, discretos y continuos con medida de puntuación LOGLIK-CG

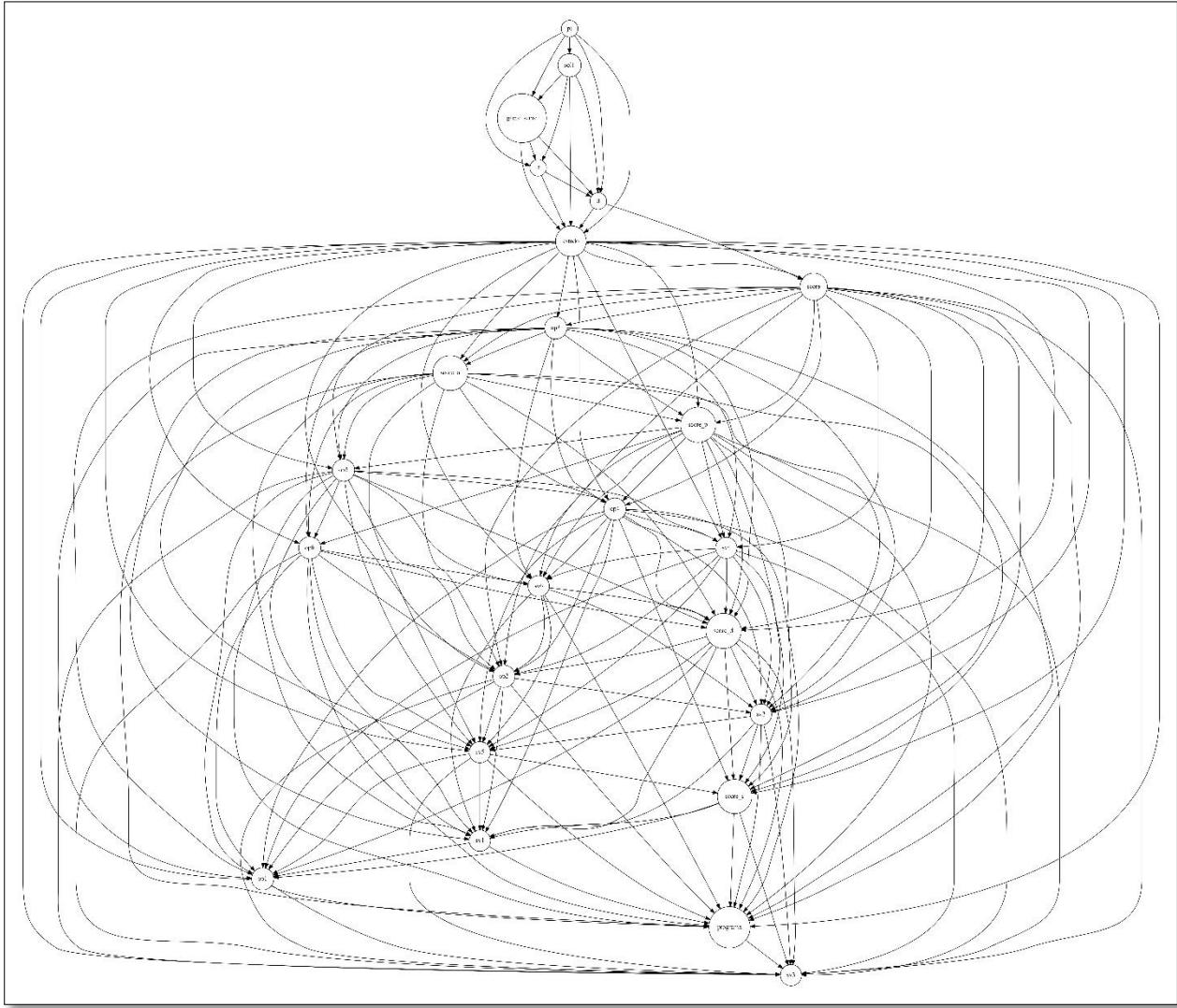


Figura 47: Red de la partición 1 del experimento con datos desbalanceados, discretos y continuos con medida de puntuación LOGLIK-CG.

9.2 DAGs de los experimentos ejecutados con Python

Se presenta una muestra de los DAGs, de la primera partición, obtenidos de todos los experimentos ejecutados con Python.

9.2.1 Experimento con datos balanceados con medida de puntuación BDEU

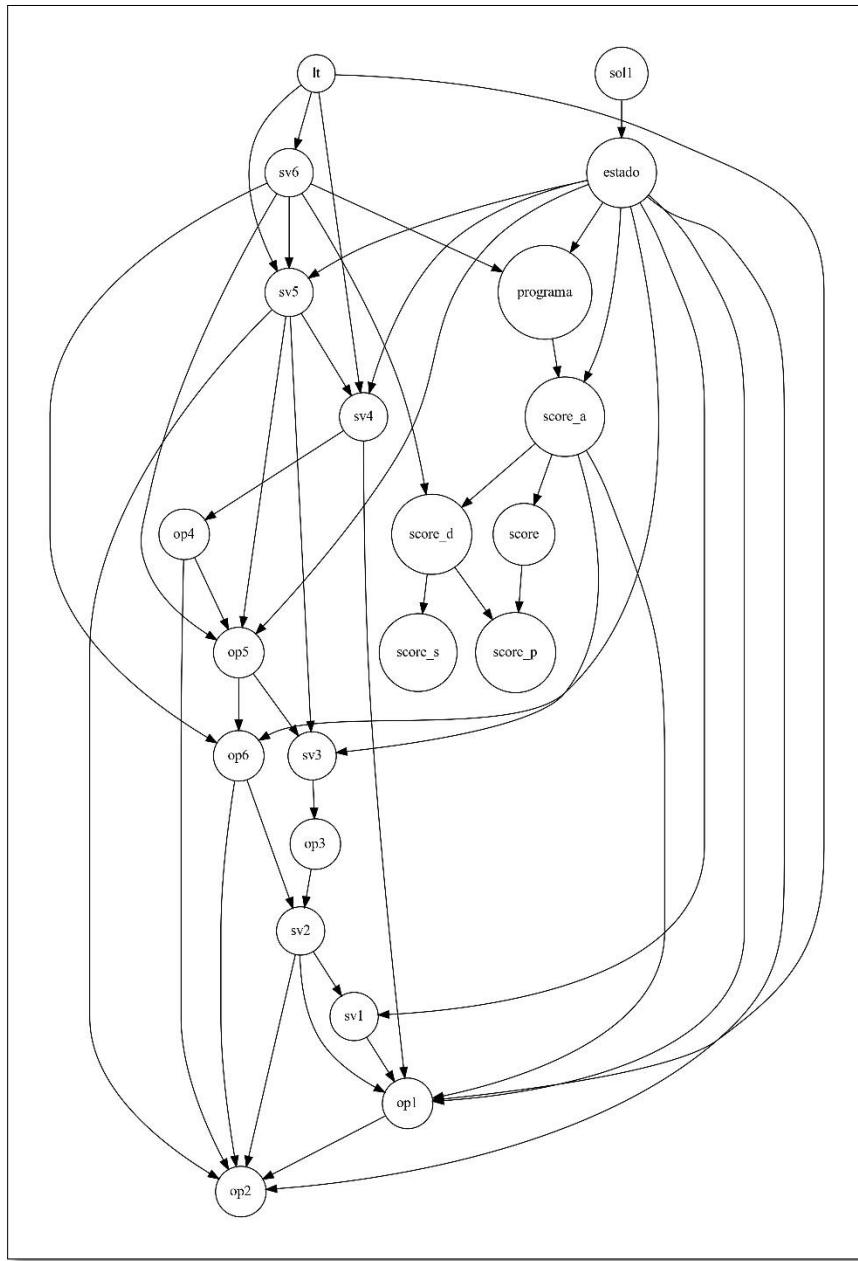


Figura 48: Red de la partición 1 del experimento con datos balanceados con medida de puntuación BDEU.

9.2.2 Experimento con datos balanceados con medida de puntuación K2

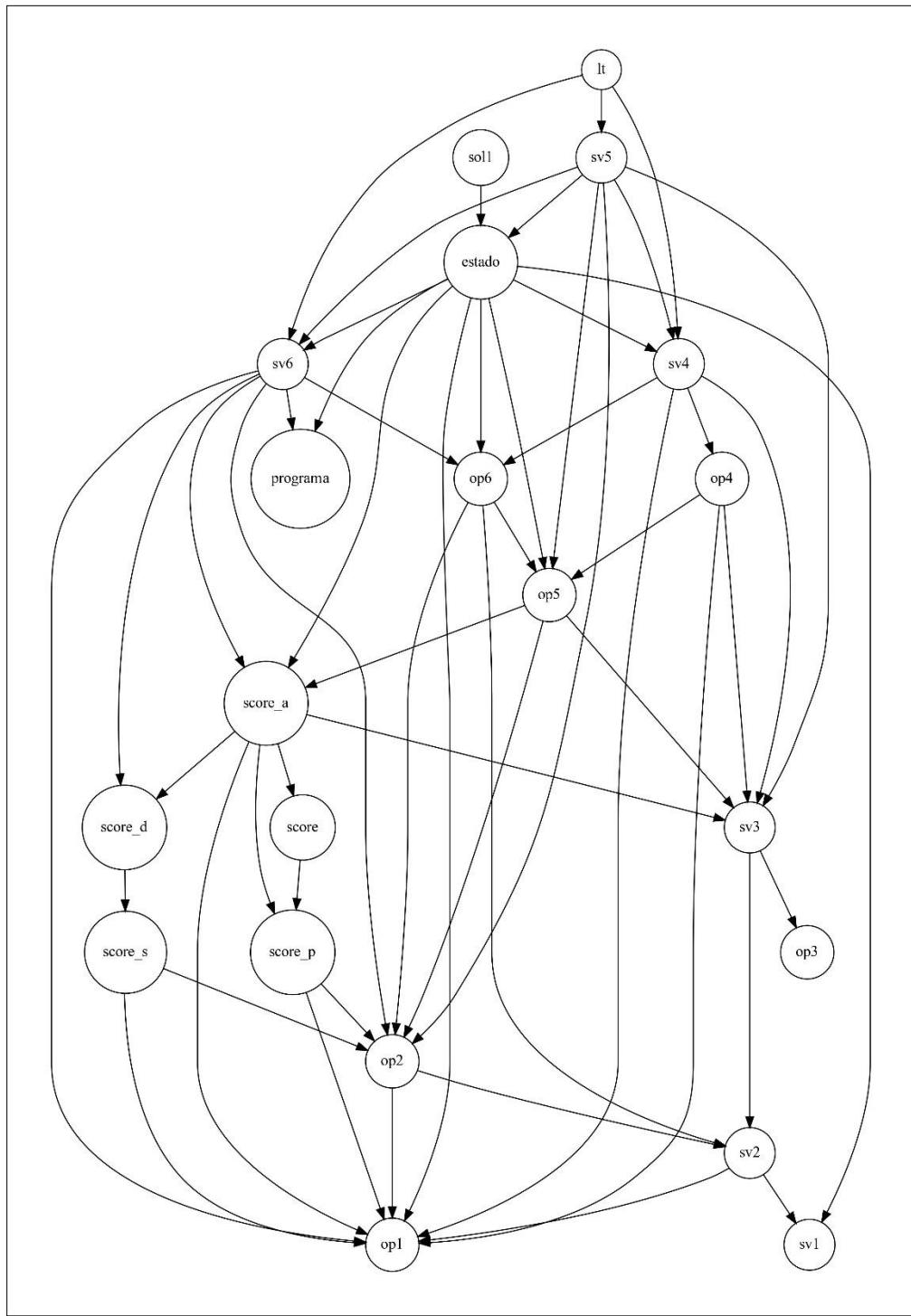


Figura 49: Red de la partición 1 del experimento con datos balanceados con medida de puntuación K2.

9.2.3 Experimento con datos balanceados con medida de puntuación BIC

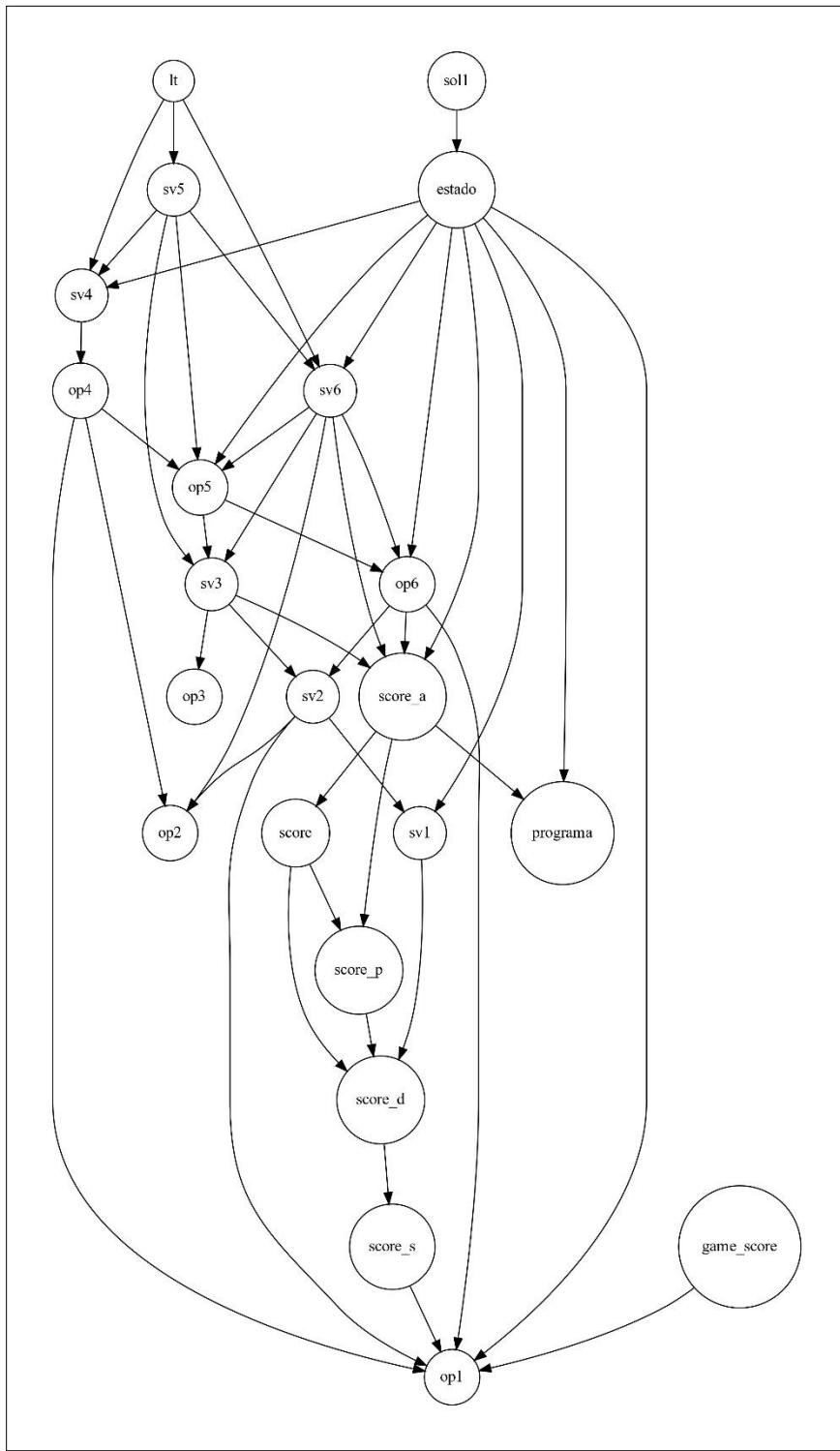


Figura 50: Red de la partición 1 del experimento con datos balanceados con medida de puntuación BIC.

9.2.4 Experimento con datos desbalanceados con medida de puntuación BDEU

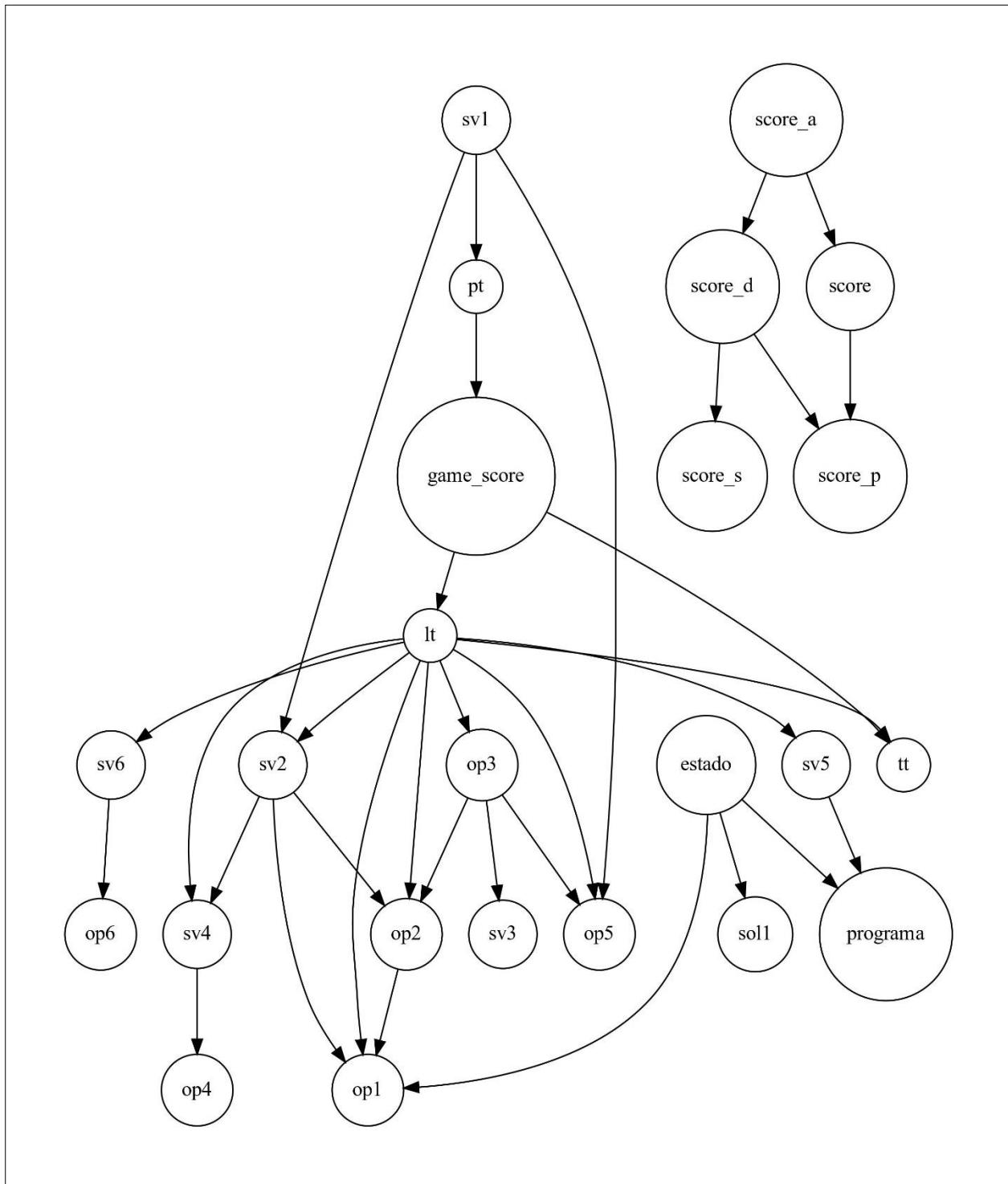


Figura 51: Red de la partición 1 del experimento con datos desbalanceados con medida de puntuación BDEU.

9.2.5 Experimento con datos desbalanceados con medida de puntuación K2

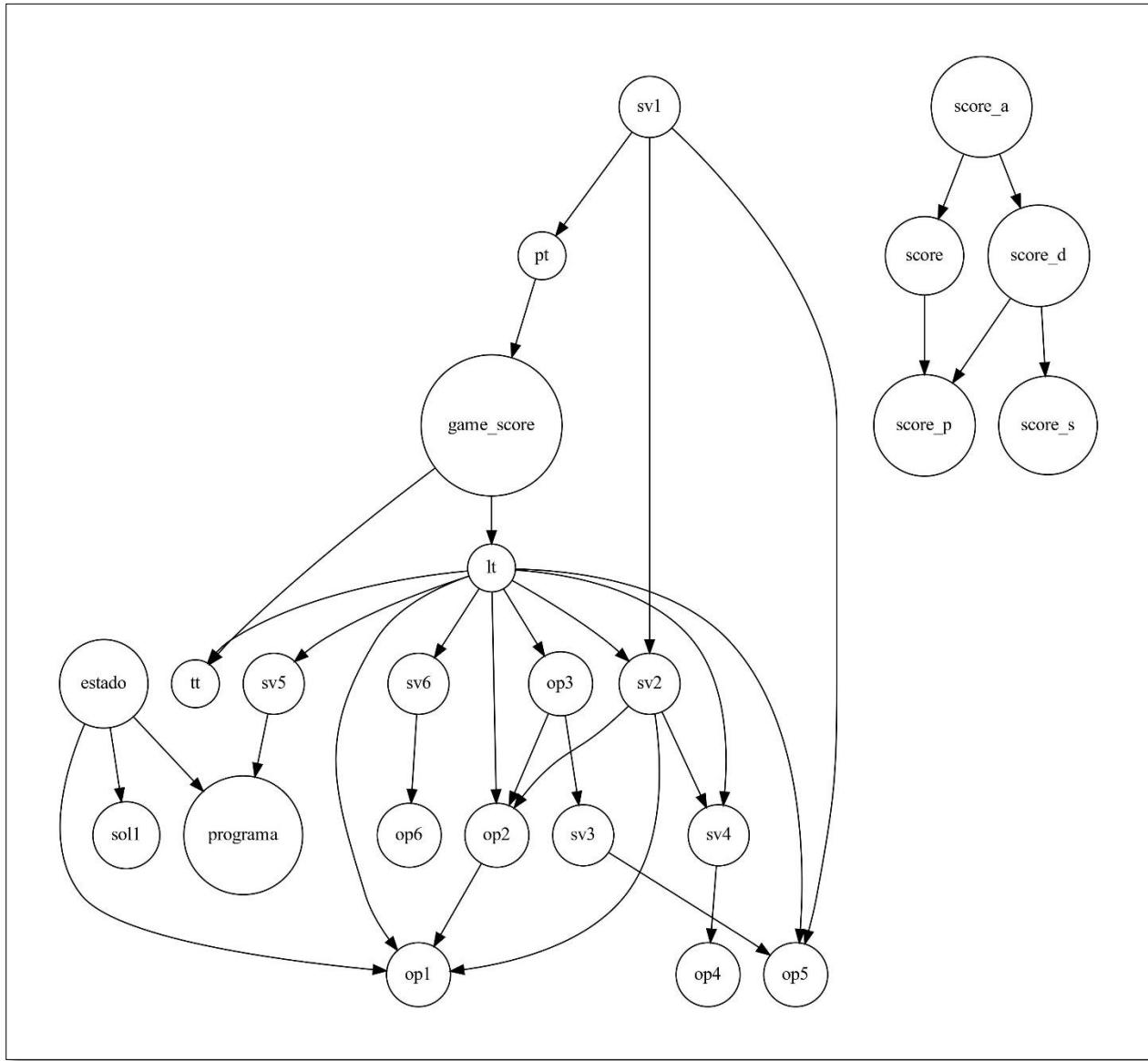


Figura 52: Red de la partición 1 del experimento con datos desbalanceados con medida de puntuación K2.

9.2.6 Experimento con datos desbalanceados con medida de puntuación BIC

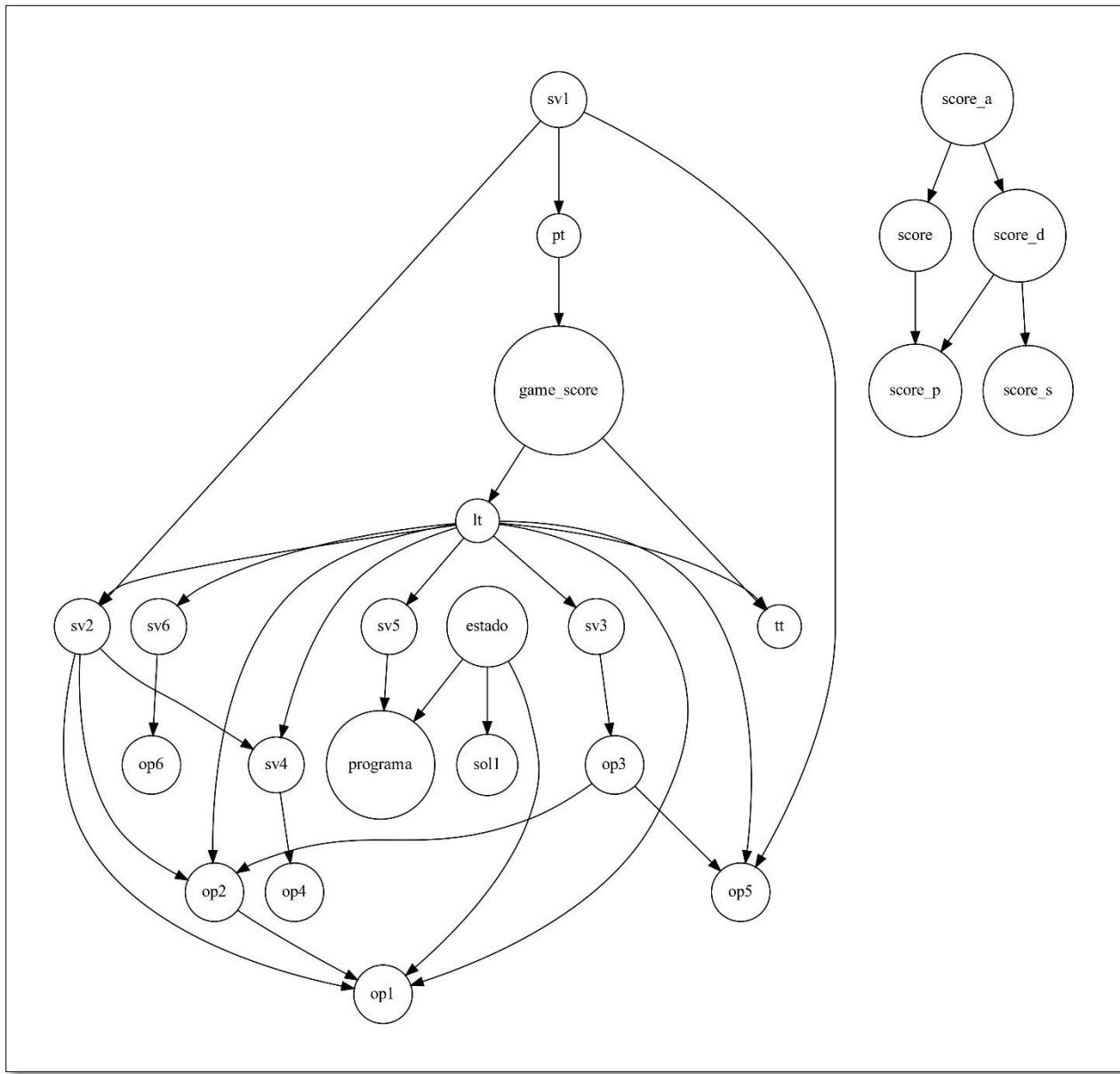


Figura 53: Red de la partición 1 del experimento con datos desbalanceados con medida de puntuación BIC.

9.3 Grafos generados en Weka

9.3.1 Experimento con variables del juego diagnóstico y algoritmo K2 Global

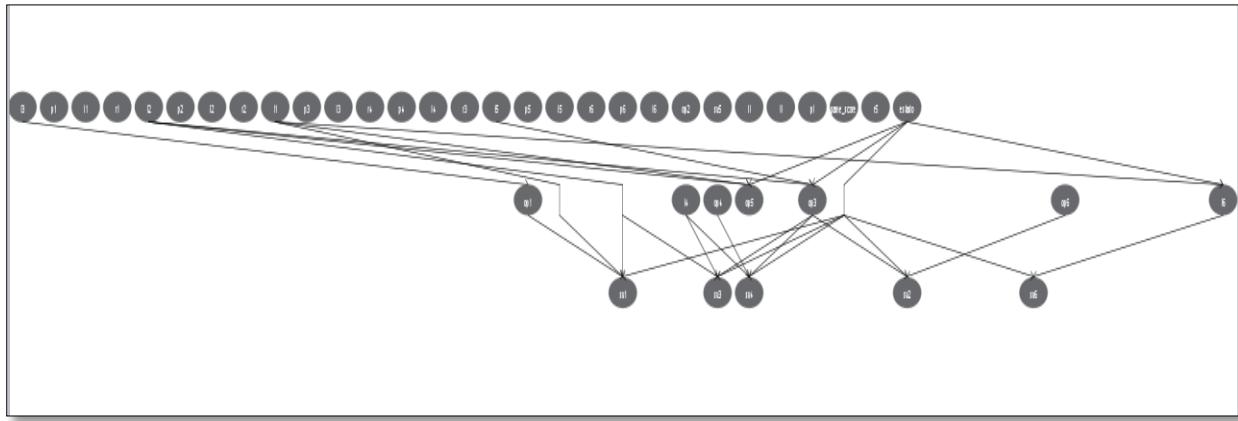


Figura 54: Red de primer experimento con datos desbalanceados y estimador de probabilidades MLE.

9.3.2 Experimento con variables del juego diagnóstico y algoritmo TabuSearch Global

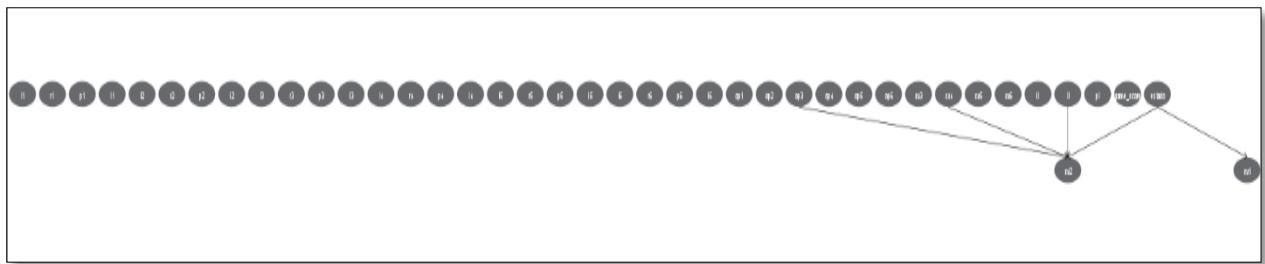


Figura 55: Red de primer experimento con datos desbalanceados y estimador de probabilidades MLE.

9.3.3 Experimento con variables del juego diagnóstico y algoritmo Repeated Hill Climber Global

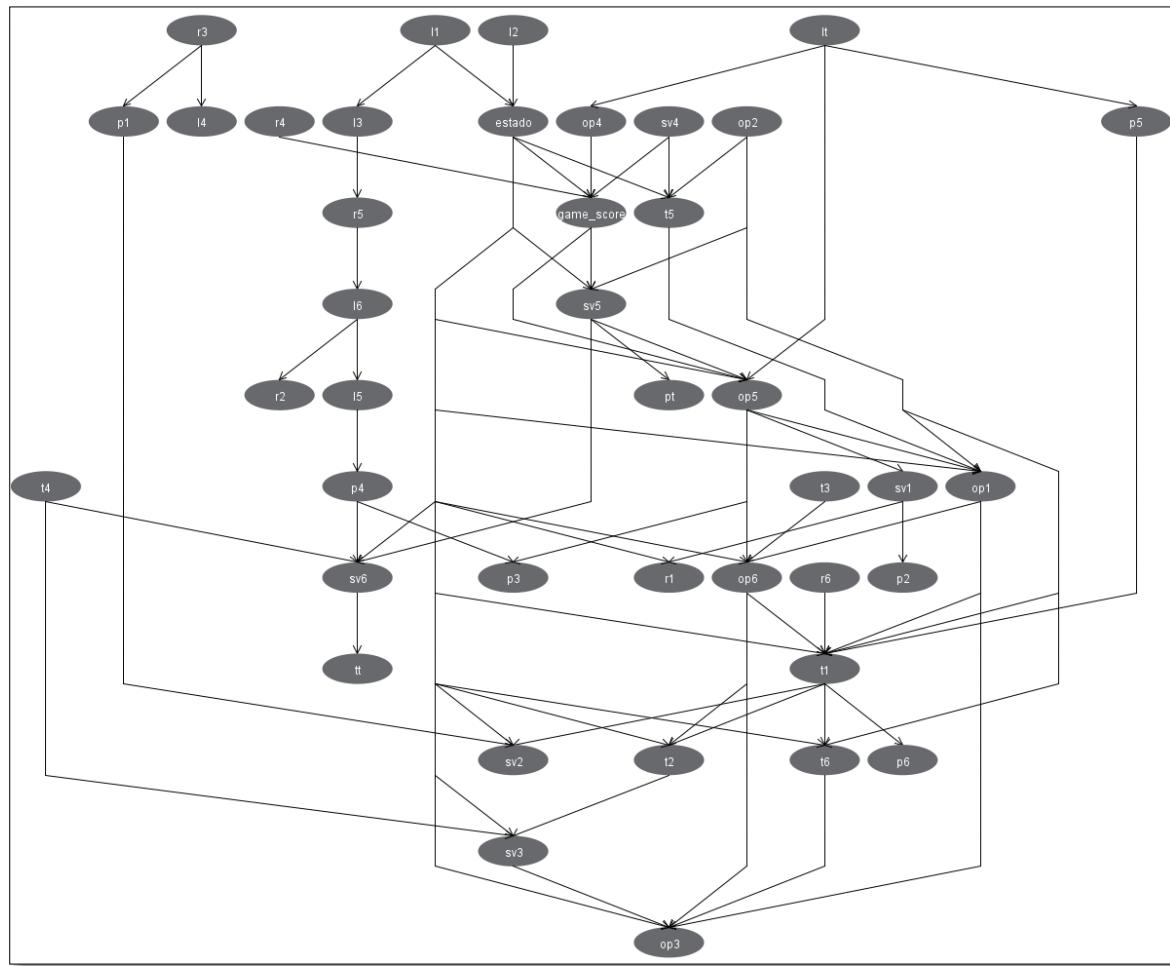


Figura 56: Red de primer experimento con datos desbalanceados y estimador de probabilidades MLE.

9.3.4 Experimento con variables del juego diagnóstico y algoritmo Hill Climber

Local

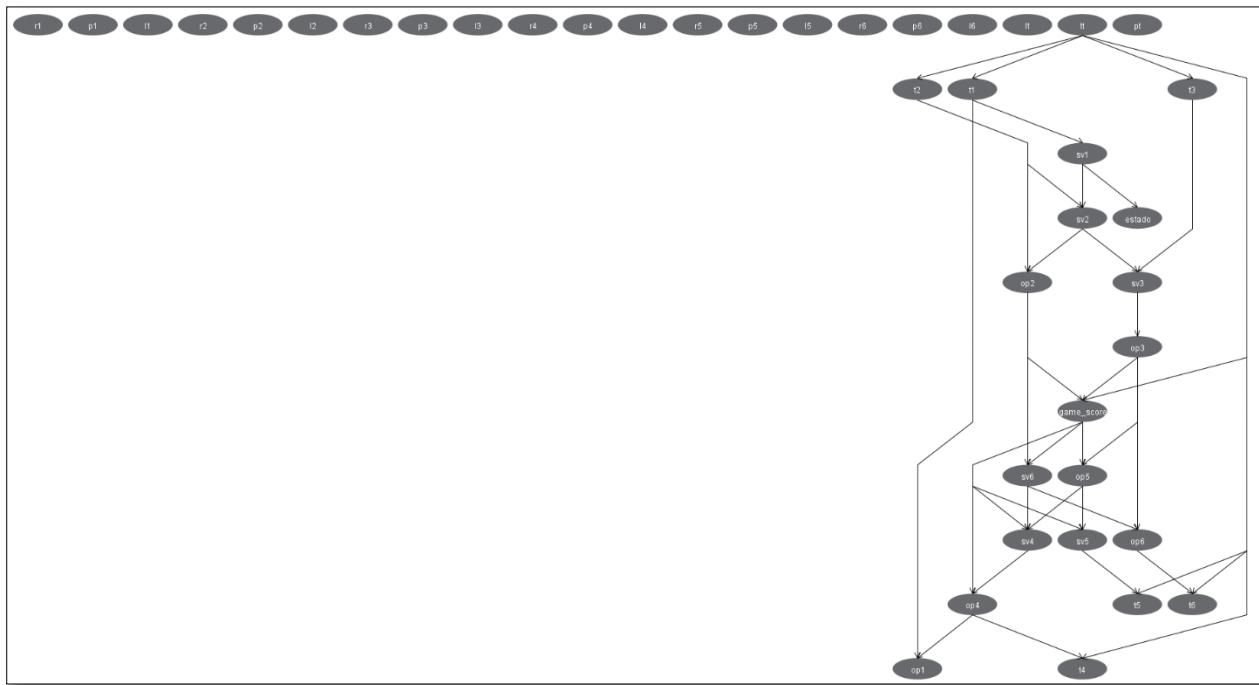


Figura 57: Red de primer experimento con datos desbalanceados, métrica AIC y estimador de probabilidades simple alfa 0.5.

9.3.5 Experimento con variables del juego diagnóstico y algoritmo Repeated Hill Climber Local

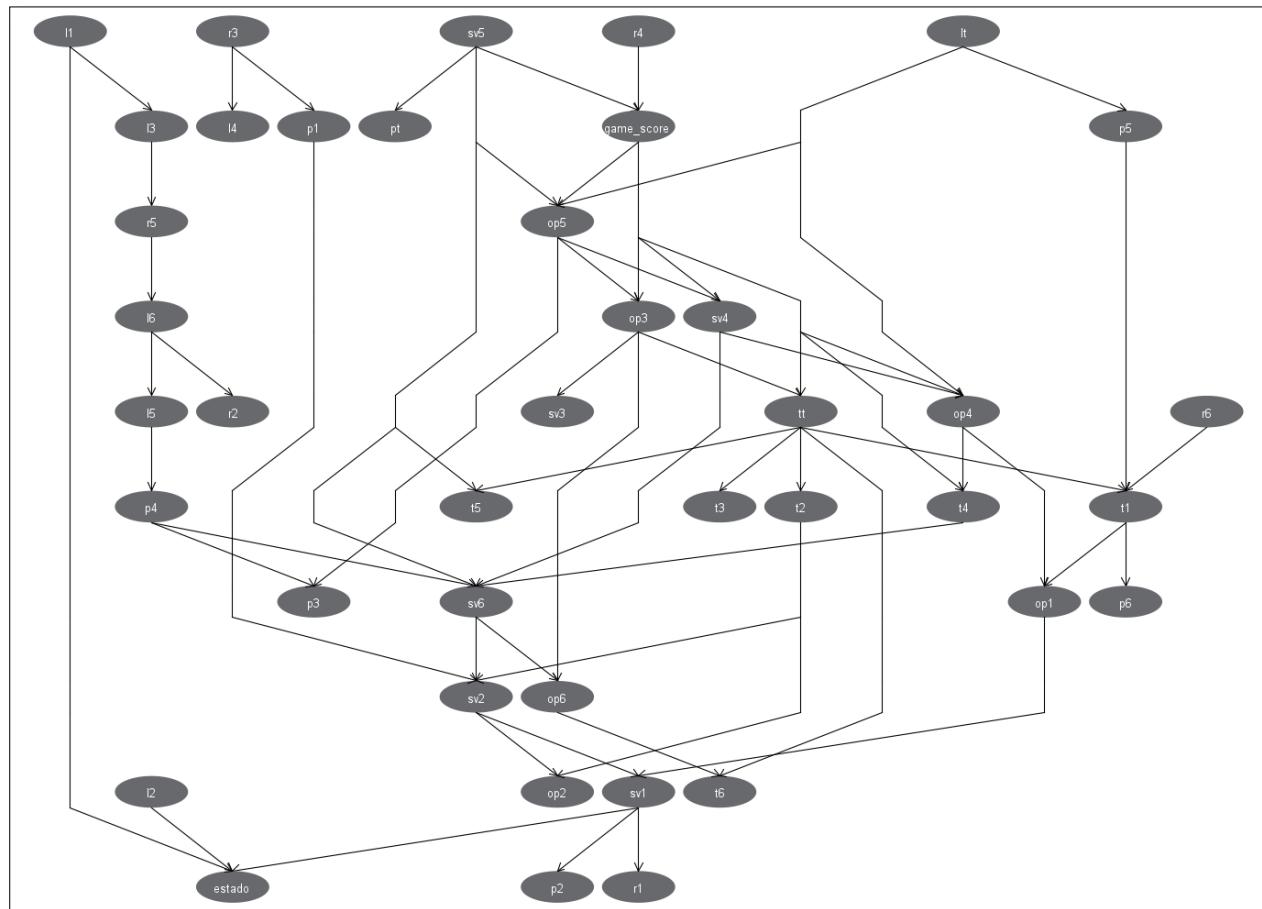


Figura 58: Red de primer experimento con datos desbalanceados, métrica AIC y estimador de probabilidades MLE

9.3.6 Experimento con variables del juego diagnóstico y algoritmo Hill Climber Global desbalanceado

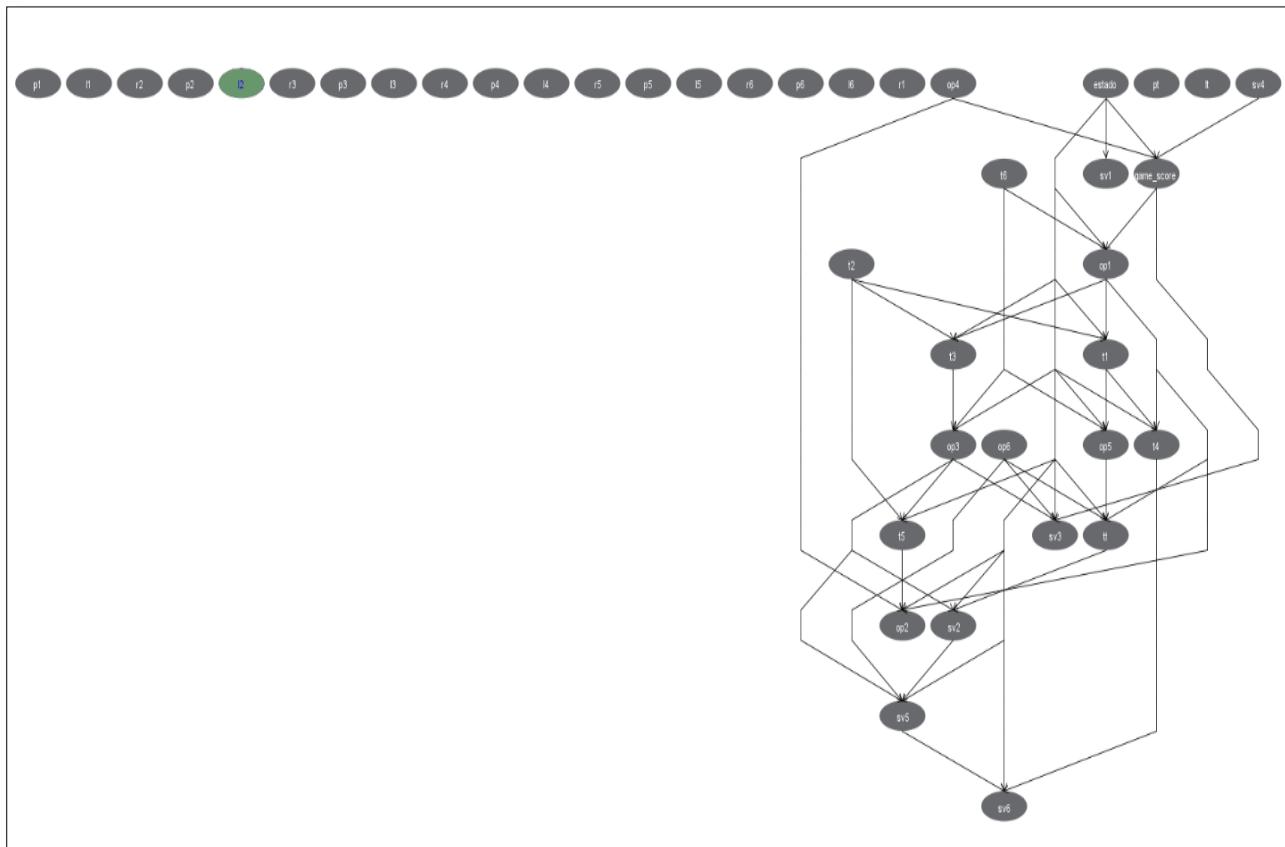


Figura 59: Red de primer experimento con datos desbalanceados y estimador de probabilidades MLE.