Instituto Tecnológico de Culiacán

Ingeniería en Sistemas Computacionales

Temas Selectos de Bases de Datos

Docente. Dr. Clemente García Gerardo

Proyecto de minería de datos

Alumno. Olan Castro Angel Eduardo

Jueves, 15 de mayo del 2022

Contenido

Generación de los datos

Con las reglas que se explicarán más abajo, diseñé un programa en Java que puede generar n ejemplos e insertarlos a la tabla de nuestra base de datos y al archivo ARFF.

Un poco del código

```
  private static Solicitante[] generarDatos(int n) {
    Solicitante[] datos = new Solicitante[n];
    for (int i = 0; i < n; i++) {
      datos[i] = generarSolicitante();
    }
    return datos;
  }

  public static Solicitante generarSolicitante() {
    int edad = Rutinas.getEdad();
    String nivelEstudios = Rutinas.getEstudios(edad);
    String nivelRenta = Rutinas.getNivelRenta(edad, nivelEstudios);
    String patrimonio = Rutinas.getPatrimonio(edad, nivelRenta);
    int numeroHijos = Rutinas.getNumeroHijos(nivelRenta);
    int tamanoCredito = Rutinas.getTamanoCredito(nivelRenta);
    String funcionario = Rutinas.getFuncionario(nivelRenta, patrimonio,
edad);
    String autorizado = Rutinas.getAutorizado(tamanoCredito, edad,
funcionario, nivelEstudios, patrimonio, nivelRenta);
    return new Solicitante(edad,
                           nivelEstudios,
                           nivelRenta,
                           patrimonio,
                           numeroHijos,
                           tamanoCredito,
                           funcionario,
                           autorizado);
  }


  public static void main(String[] args) {
    Solicitante[] datos = generarDatos(100000);
    poblarTablaSQL(datos);
    generarARFF(datos);
  }
```

Donde el solicitante es una abstracción para representar y guardar fácilmente cada ejemplo en la memoria, su código java es el siguiente, ignorando los getters:

```
public class Solicitante {
  private String nivelEstudios, nivelRenta, patrimonio,
                 funcionario, autorizado;
  private int edad, numeroHijos, tamanoCredito;
  public Solicitante(int edad, String nivelEstudios, String nivelRenta,
String patrimonio,int numeroHijos, int tamanoCredito, String funcionario,
String autorizado) {
    this.edad = edad;
    this.nivelEstudios = nivelEstudios;
    this.nivelRenta = nivelRenta;
    this.patrimonio = patrimonio;
    this.numeroHijos = numeroHijos;
    this.tamanoCredito = tamanoCredito;
    this.funcionario = funcionario;
    this.autorizado = autorizado;
  }

  @Override
  public String toString() {
    return edad + "," + nivelEstudios + "," + nivelRenta + "," + patrimonio
+ "," + numeroHijos + "," + tamanoCredito + "," + funcionario + "," +
autorizado + "\n";
  }
}
```

Reglas para darle coherencia a los datos

Para que los datos tengan sentido he decidido combinar algunas de las restricciones impuestas por el profesor con otras mías, intentando que estos se apeguen lo mayor posible a la realidad, para denotar estas reglas, usé una especie de árbol para elegir el valor que cada atributo será seleccionado en función de otros y con cierta probabilidad. Decidí esa notación porque me parece fácil de leer, tal como si fuera un código, retornando cuando se presentan los porcentajes.

Por ejemplo, para el nivel de renta, se debe de leer como: si la edad es menor o igual a 40, entonces, si se tiene licenciatura o posgrado, este tiene un 30% de probabilidad de tener baja renta, un 60% de media y 10% de alta, en cambio, si no tiene estudios mayores, este tiene una probabilidad del 40% de tener una renta baja, 50% de media y 5% de alta. Retrocediendo un poco, si la edad es mayor a 40, se aplica una lógica similar.

Edad: número aleatorio en el rango 20-60

Nivel de estudios:

Edad <= 23: 1% posgrado, 79% licenciatura, 20% ninguno

Edad <= 40: 5% posgrado, 65% licenciatura, 30% ninguno

Edad > 40: 2% posgrado, 28% licenciatura, 70% ninguno

Nivel de renta:

Edad <= 40:

Licenciatura o posgrado: 30% bajo, 60% medio, 10% alto

Sin estudios: 40% bajo, 50% medio, 5% alto

Edad > 40:

Licenciatura o posgrado: 10% bajo, 65% medio, 25% alto

Sin estudios: 78% bajo, 20% medio, 2% alto

Patrimonio:

Edad <= 35:

Nivel de renta alto o medio: 30% bajo, 35% medio, 35% alto

Nivel de renta bajo: 75% bajo, 23% medio, 2% alto

Edad > 35:

Nivel de renta alto o medio: 20% bajo, 50% medio, 30% alto

Nivel de renta bajo: 90% bajo, 9% medio, 1% alto

Número de hijos:

Nivel de renta bajo: 1-6 hijos con misma probabilidad

Nivel de renta medio o alto: 0-3 hijos con misma probabilidad

Tamaño del crédito:

Nivel de renta bajo: entero en el rango 500-20,000

Nivel de renta medio o alto: entero en el rango 500-100,000

Funcionario:

Nivel de renta bajo:

Patrimonio bajo: 80% no y 20% si

Patrimonio medio o alto:

Edad <= 25: 25% no y 75% si

Edad > 25: 40% no y 60% si

Nivel de renta medio o alto: 10% no y 90% si

Autorizó:

Credito <= 30,000:

No es funcionario:

Edad >= 55: 95% no y 5% si

Patrimonio bajo o medio: 80% no y 20% si

Patrimonio alto: 50% no y 50% si

Sin estudios:

Nivel de renta bajo: 65% no y 35% si

Nivel de renta medio o alto: 30% no y 70% si

Licenciatura o posgrado:

Nivel de renta bajo: 60% no y 40% si

Nivel de renta medio o alto: 15% no y 85% si

Credito <= 60,000:

No es funcionario:

Edad >= 50: 95% no y 5% si

Nivel de renta alto: 25% no y 75% si

Patrimonio alto: 20% no y 80% si

Si no: 70% no y 30% si

Nivel de renta medio o alto: 10% no y 90% si

Licenciatura o posgrado: 40% no y 60% si

Si no: 80% no y 20% si

Credito <= 100,000:

No es funcionario:

Edad >= 40: 95% no y 5% si

Nivel de renta alto: 35% no y 65% si

Patrimonio alto: 30% no y 60% si

Si no: 80% no y 20% si

Nivel de renta alto: 10% no y 90% si

Nivel de renta medio: 25% no y 75% si

Licenciatura o posgrado: 60% no y 40% si

Si no: 90% no y 10% si

Para lograr esto, en una clase Rutinas.java implementé todos los métodos necesarios, para aplicar las probabilidades, por ejemplo, sobre 100, generé un número aleatorio entero en el rango 1-100 y dividí el intervalo 1-100 en tantos intervalos como distintos caminos haya, si se tiene 60%, 30% y 10%, entonces hay 3 intervalos: 1-60, 61-90 y 91-100, basta con ver en qué cayó el número aleatorio para asignar un valor terminal.

Código:

// Las constantes son para reducir los bugs, realmente son Strings.

```
import java.util.Random;

public class Rutinas {
  private static Random rnd = new Random();

  public static int getEdad() {
    return rnd.nextInt(20, 61);
  }

  public static String getEstudios(int edad) {
    int r = rnd.nextInt(1, 101);
    if (edad <= 23) {
      if (r == 1) {
        return Constantes.POSGRADO;
      }
```

```
      if (r <= 80) {
        return Constantes.LICENCIATURA;
      }
      return Constantes.NINGUNO;
    }
    if (edad <= 40) {
      if (r <= 5) {
        return Constantes.POSGRADO;
      }
      if (r <= 70) {
        return Constantes.LICENCIATURA;
      }
      return Constantes.NINGUNO;
    }
    if (r <= 2) {
      return Constantes.POSGRADO;
    }
    if (r <= 30) {
      return Constantes.POSGRADO;
    }
    return Constantes.NINGUNO;
  }

  public static String getNivelRenta(int edad, String nivelEstudios) {
    int r = rnd.nextInt(1, 101);
    if (edad <= 40) {
      if (nivelEstudios.equals(Constantes.LICENCIATURA) ||
nivelEstudios.equals(Constantes.POSGRADO)) {
        if (r <= 30) {
          return Constantes.BAJO;
        }
        if (r <= 90) {
          return Constantes.MEDIO;
        }
        return Constantes.ALTO;
      }
      if (r <= 40) {
        return Constantes.BAJO;
      }
      if (r <= 95) {
        return Constantes.MEDIO;
      }
      return Constantes.ALTO;
    }
```

```
    if (nivelEstudios.equals(Constantes.LICENCIATURA) ||
nivelEstudios.equals(Constantes.POSGRADO)) {
      if (r <= 10) {
        return Constantes.BAJO;
      }
      if (r <= 75) {
        return Constantes.MEDIO;
      }
      return Constantes.ALTO;
    }
    if (r <= 78) {
      return Constantes.BAJO;
    }
    if (r <= 98) {
      return Constantes.MEDIO;
    }
    return Constantes.ALTO;
  }

  public static String getPatrimonio(int edad, String nivelRenta) {
    int r = rnd.nextInt(1, 101);
    if (edad <= 35) {
      if (nivelRenta.equals(Constantes.ALTO) ||
nivelRenta.equals(Constantes.MEDIO)) {
        if (r <= 30) {
          return Constantes.BAJO;
        }
        if (r <= 65) {
          return Constantes.MEDIO;
        }
        return Constantes.ALTO;
      }
      if (r <= 75) {
        return Constantes.BAJO;
      }
      if (r <= 98) {
        return Constantes.MEDIO;
      }
      return Constantes.ALTO;
    }
    if (nivelRenta.equals(Constantes.ALTO) ||
nivelRenta.equals(Constantes.MEDIO)) {
      if (r <= 20) {
        return Constantes.BAJO;
      }
```

```
      if (r <= 70) {
        return Constantes.MEDIO;
      }
      return Constantes.ALTO;
    }
    if (r <= 90) {
      return Constantes.BAJO;
    }
    if (r <= 99) {
      return Constantes.MEDIO;
    }
    return Constantes.ALTO;
  }


  public static int getNumeroHijos(String nivelRenta) {
    if (nivelRenta.equals(Constantes.BAJO)) {
      return rnd.nextInt(1, 7);
    }
    return rnd.nextInt(0, 4);
  }

  public static int getTamanoCredito(String nivelRenta) {
    if (nivelRenta.equals(Constantes.BAJO)) {
      return rnd.nextInt(500, 20000);
    }
    return rnd.nextInt(500, 100000);
  }

  public static String getFuncionario(String nivelRenta, String patrimonio,
int edad) {
    int r = rnd.nextInt(1, 101);
    if (nivelRenta.equals(Constantes.BAJO)) {
      if (patrimonio.equals(Constantes.BAJO)) {
        return decisionBinaria(r, 80);
      }
      if (edad <= 25) {
        return decisionBinaria(r, 25);
      }
      return decisionBinaria(r, 40);
    }
    return decisionBinaria(r, 10);
  }
```

```
  public static String getAutorizado(int credito, int edad, String
funcionario,
                                     String nivelEstudios, String
patrimonio, String nivelRenta) {
    int r = rnd.nextInt(1, 101);
    if (credito <= 30000) {
      if (funcionario.equals(Constantes.NO)) {
        if (edad >= 55) {
          return decisionBinaria(r, 95);
        }
        if (patrimonio.equals(Constantes.BAJO) ||
patrimonio.equals(Constantes.MEDIO)) {
          return decisionBinaria(r, 80);
        }
        return decisionBinaria(r, 50);
      }
      if (nivelEstudios.equals(Constantes.NINGUNO)) {
        if (nivelRenta.equals(Constantes.BAJO)) {
          return decisionBinaria(r, 65);
        }
        return decisionBinaria(r, 30);
      }
      if (nivelRenta.equals(Constantes.BAJO)) {
        return decisionBinaria(r, 60);
      }
      return decisionBinaria(r, 15);
    }
    if (credito <= 60000) {
      if (funcionario.equals(Constantes.NO)) {
        if (edad >= 50) {
          return decisionBinaria(r, 95);
        }
        if (nivelRenta.equals(Constantes.ALTO)) {
          return decisionBinaria(r, 25);
        }
        if (patrimonio.equals(Constantes.ALTO)) {
          return decisionBinaria(r, 20);
        }
        return decisionBinaria(r, 70);
      }
      if (nivelRenta.equals(Constantes.MEDIO) ||
nivelRenta.equals(Constantes.ALTO)) {
        return decisionBinaria(r, 10);
      }
```

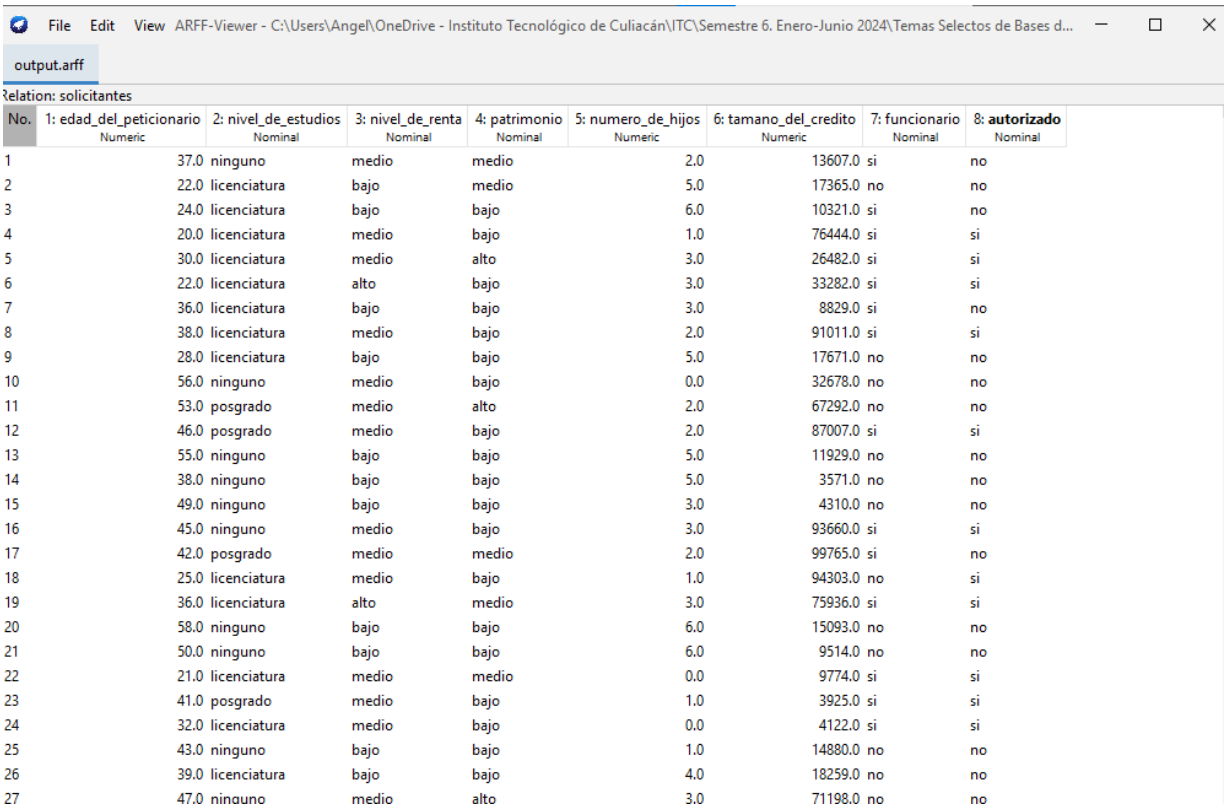
```
      if (nivelEstudios.equals(Constantes.LICENCIATURA) ||
nivelEstudios.equals(Constantes.POSGRADO)) {
        return decisionBinaria(r, 40);
      }
      return decisionBinaria(r, 80);
    }
    if (funcionario.equals(Constantes.NO)) {
      if (edad >= 40) {
        return decisionBinaria(r, 95);
      }
      if (nivelRenta.equals(Constantes.ALTO)) {
        return decisionBinaria(r, 35);
      }
      if (patrimonio.equals(Constantes.ALTO)) {
        return decisionBinaria(r, 30);
      }
      return decisionBinaria(r, 80);
    }
    if (nivelRenta.equals(Constantes.ALTO)) {
      return decisionBinaria(r, 10);
    }
    if (nivelRenta.equals(Constantes.MEDIO)) {
      return decisionBinaria(r, 25);
    }
    if (nivelEstudios.equals(Constantes.LICENCIATURA) ||
nivelEstudios.equals(Constantes.POSGRADO)) {
      return decisionBinaria(r, 60);
    }
    return decisionBinaria(r, 90);
  }

  private static String decisionBinaria(int valor, int porcentajeNo) {
    return valor <= porcentajeNo ? Constantes.NO : Constantes.SI;
  }
}
```

Archivo ARFF

El método es bastante simple, borramos el contenido del archivo ARFF (aunque podríamos no hacerlo fácilmente), escribimos su encabezado típico y posteriormente cada uno de los ejemplos aleatoriamente calculados:

```
  private static void generarARFF(Solicitante[] datos) {
    try (BufferedWriter bufferedWriter = new BufferedWriter(new
FileWriter("src/output.arff"))) {
      llenarEncabezadoARFF(bufferedWriter);
      for (Solicitante solicitante : datos) {
        bufferedWriter.write(solicitante.toString());
      }
    } catch (IOException e) {
      e.printStackTrace();
    }
    System.out.println("Escritura en ARFF terminada");
  }

  private static void llenarEncabezadoARFF(BufferedWriter file) {
    String encabezado = "@relation solicitantes\n" +
                        "\n" +
                        "@attribute edad_del_peticionario real\n" +
                        "@attribute nivel_de_estudios {ninguno,
licenciatura, posgrado}\n" +
                        "@attribute nivel_de_renta {alto, medio, bajo}\n" +
                        "@attribute patrimonio {alto, medio, bajo}\n" +
                        "@attribute numero_de_hijos real\n" +
                        "@attribute tamano_del_credito real\n" +
                        "@attribute funcionario {si, no}\n" +
                        "@attribute autorizado {si, no}\n" +
                        "\n" +
                        "@data\n";
    try {
      file.write(encabezado);
    } catch (IOException e) {
      // TODO Auto-generated catch block
      e.printStackTrace();
    }
  }
```

Una muestra de todos los ejemplos en el archivo ARFF, desde el visualizador de ARFF integrado en WEKA:



No.	1: edad_del_peticionario Numeric	2: nivel_de_estudios Nominal	3: nivel_de_renta Nominal	4: patrimonio Nominal	5: numero_de_hijos Numeric	6: tamano_del_credito Numeric	7: funcionario Nominal	8: **autorizado** Nominal
1	37.0	ninguno	medio	medio	2.0	13607.0	si	no
2	22.0	licenciatura	bajo	medio	5.0	17365.0	no	no
3	24.0	licenciatura	bajo	bajo	6.0	10321.0	si	no
4	20.0	licenciatura	medio	bajo	1.0	76444.0	si	si
5	30.0	licenciatura	medio	alto	3.0	26482.0	si	si
6	22.0	licenciatura	alto	bajo	3.0	33282.0	si	si
7	36.0	licenciatura	bajo	bajo	3.0	8829.0	si	no
8	38.0	licenciatura	medio	bajo	2.0	91011.0	si	si
9	28.0	licenciatura	bajo	bajo	5.0	17671.0	no	no
10	56.0	ninguno	medio	bajo	0.0	32678.0	no	no
11	53.0	posgrado	medio	alto	2.0	67292.0	no	no
12	46.0	posgrado	medio	bajo	2.0	87007.0	si	si
13	55.0	ninguno	bajo	bajo	5.0	11929.0	no	no
14	38.0	ninguno	bajo	bajo	5.0	3571.0	no	no
15	49.0	ninguno	bajo	bajo	3.0	4310.0	no	no
16	45.0	ninguno	medio	bajo	3.0	93660.0	si	si
17	42.0	posgrado	medio	medio	2.0	99765.0	si	no
18	25.0	licenciatura	medio	bajo	1.0	94303.0	no	si
19	36.0	licenciatura	alto	medio	3.0	75936.0	si	si
20	58.0	ninguno	bajo	bajo	6.0	15093.0	no	no
21	50.0	ninguno	bajo	bajo	6.0	9514.0	no	no
22	21.0	licenciatura	medio	medio	0.0	9774.0	si	si
23	41.0	posgrado	medio	bajo	1.0	3925.0	si	si
24	32.0	licenciatura	medio	bajo	0.0	4122.0	si	si
25	43.0	ninguno	bajo	bajo	1.0	14880.0	no	no
26	39.0	licenciatura	bajo	bajo	4.0	18259.0	no	no
27	47.0	ninguno	medio	alto	3.0	71198.0	no	no

Tabla de la base de datos

El esquema de la base de datos y la tabla se definió de la siguiente manera:

```
CREATE DATABASE Mineros
GO
USE Mineros
GO
CREATE TABLE Solicitantes(
	EdadDelPeticionario TINYINT,
	NivelDeEstudios VARCHAR(12),
	NivelDeRenta  VARCHAR(5),
	Patrimonio VARCHAR(5),
	NumeroDeHijos TINYINT,
	TamanoDelCredito INT,
	Funcionario CHAR(2),
	Autorizado CHAR(2)
)
select * from solicitantes
```

Ahora, ya desde el código en java, toda la interacción con la base de datos fue por medio de una instancia de una clase BaseDeDatos, definida de la siguiente manera:

```
import java.sql.Statement;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.PreparedStatement;
import java.sql.SQLException;

public class BaseDeDatos {
  private Connection connection;
  private PreparedStatement pst;
  private int cnt = 0;

  public BaseDeDatos(String url, String user, String password) {
    try {
      connection = DriverManager.getConnection(url, user, password);
      pst = connection.prepareStatement("INSERT INTO SOLICITANTES
VALUES(?,?,?,?,?,?,?,?)");
      System.out.println("Conexión establecida correctamente");
    }
    catch (Exception e) {
      System.err.println(e.getMessage());
    }
  }

  public void limpiarTabla() {
    try {
```

```
      connection.createStatement().executeUpdate("DELETE FROM
SOLICITANTES");
    } catch (SQLException e) {
      // TODO Auto-generated catch block
      e.printStackTrace();
    }
  }

  public void insertarSolicitante(Solicitante solicitante) {
    try {
      pst.setInt(1, solicitante.getEdad());
      pst.setString(2, solicitante.getNivelEstudios());
      pst.setString(3, solicitante.getNivelRenta());
      pst.setString(4, solicitante.getPatrimonio());
      pst.setInt(5, solicitante.getNumeroHijos());
      pst.setInt(6, solicitante.getTamanoCredito());
      pst.setString(7, solicitante.getFuncionario());
      pst.setString(8, solicitante.getFuncionario());

      pst.addBatch();
    } catch (SQLException e) {
      // TODO Auto-generated catch block
      e.printStackTrace();
    }
  }

  public void ejecutarActualizacion() {
    try {
      pst.executeBatch();
    } catch (SQLException e) {
      // TODO Auto-generated catch block
      e.printStackTrace();
    }
  }

  public void cerrarConexion() {
    try {
      connection.close();
    } catch (Exception e) {
      // TODO: handle exception
    }
  }
}
```

Desde la clase principal:

```
  private static void poblarTablaSQL(Solicitante[] datos) {
    BaseDeDatos bd = new
BaseDeDatos("jdbc:sqlserver://localhost:1433;rewriteBatchedStatements=true;d
atabaseName=Mineros;trustServerCertificate=true;loginTimeOut=3", "sa",
"sa");
    bd.limpiarTabla();
    for (Solicitante solicitante : datos) {
      bd.insertarSolicitante(solicitante);
    }
    bd.ejecutarActualizacion();
    bd.cerrarConexion();
    System.out.println("SQL Server terminado");
  }
```

Nótese que en algunas partes del código se puede notar la palabra “Batch”. Desde el código se realiza una cantidad enorme de inserciones y enviar una por una las sentencias a la base de datos pueden resultar muy costoso en cuestión de tiempo, para mejorar eso, intenté darle a la interacción con la base de datos un enfoque de procesamiento en lotes. Básicamente, en este contexto, el procesamiento en lotes nos permite almacenar temporalmente las sentencias SQL y en algún punto (en este caso, al final), enviar todas las del lote juntas al servidor de SQL Server para que las ejecute, siendo mucho más eficiente, pues me toma unos 10 segundos insertar 100k de tuplas.

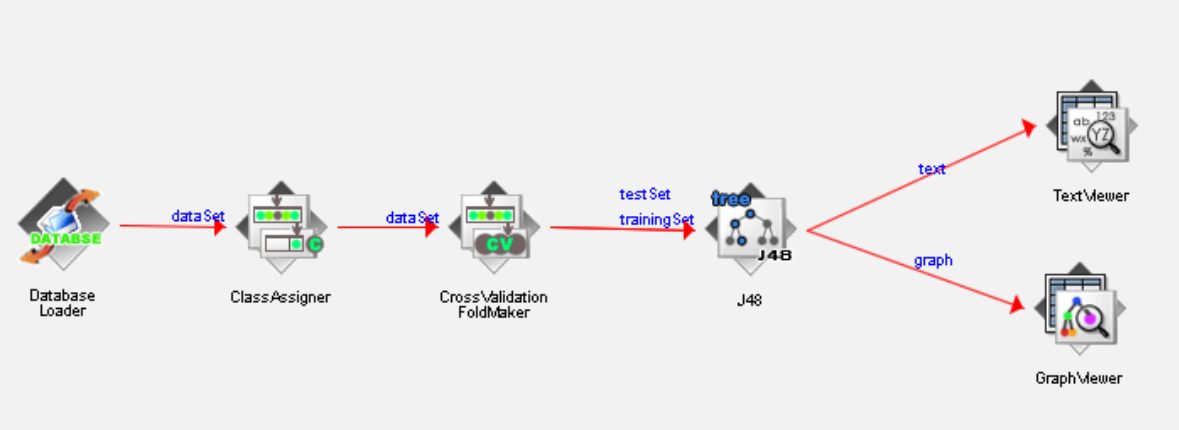
Algunas tuplas de la tabla:

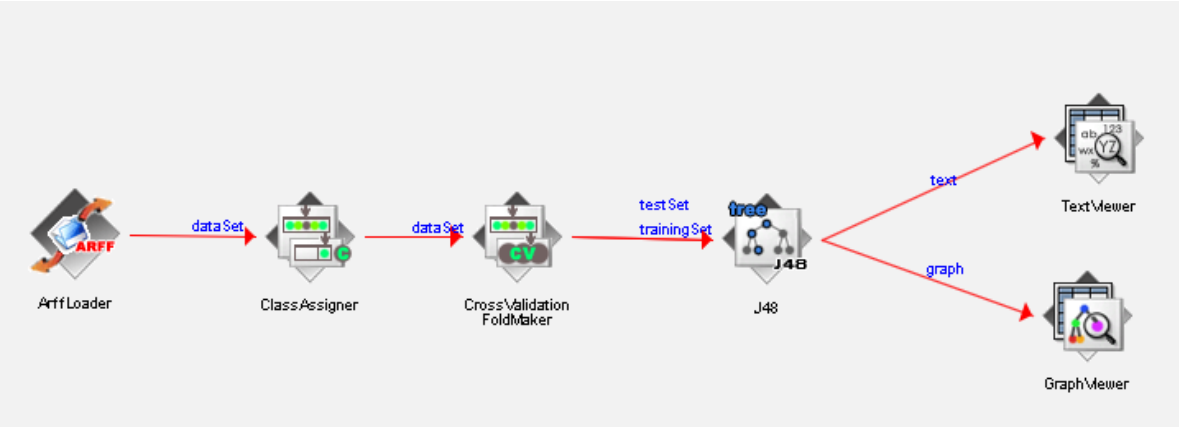
	EdadDelPeticionario	NivelDeEstudios	NivelDeRenta	Patrimonio	NumeroDeHijos	TamanoDelCredito	Funcionario	Autorizado
1	43	ninguno	medio	alto	0	55465	si	si
2	41	ninguno	bajo	bajo	1	7378	si	si
3	45	ninguno	bajo	medio	2	8044	si	si
4	48	ninguno	medio	alto	1	85703	si	si
5	29	licenciatura	medio	medio	2	55972	si	si
6	40	licenciatura	bajo	medio	5	7316	si	si
7	29	ninguno	medio	medio	2	49670	si	si
8	28	ninguno	bajo	bajo	6	1868	si	si
9	36	licenciatura	bajo	bajo	2	3402	no	no
10	23	licenciatura	medio	medio	0	19894	si	si
11	35	posgrado	medio	alto	0	41585	si	si
12	24	ninguno	bajo	medio	1	753	si	si
13	26	licenciatura	bajo	bajo	6	19459	no	no
14	54	ninguno	medio	medio	3	68952	si	si
15	21	licenciatura	medio	bajo	2	29634	si	si
16	40	ninguno	bajo	bajo	3	12778	no	no
17	33	licenciatura	medio	medio	2	92385	si	si
18	27	licenciatura	bajo	alto	3	14488	si	si
19	45	ninguno	medio	alto	0	19307	si	si
20	45	posgrado	medio	bajo	0	27131	si	si
21	47	posgrado	alto	bajo	1	86354	si	si
22	27	licenciatura	medio	alto	2	73256	si	si

Minería de datos

Flujos de conocimiento

Tenemos 2 flujos en total, siendo el primero el que toma como entrada la tabla de la base de datos y el segundo el que se alimenta del archivo ARFF generado.

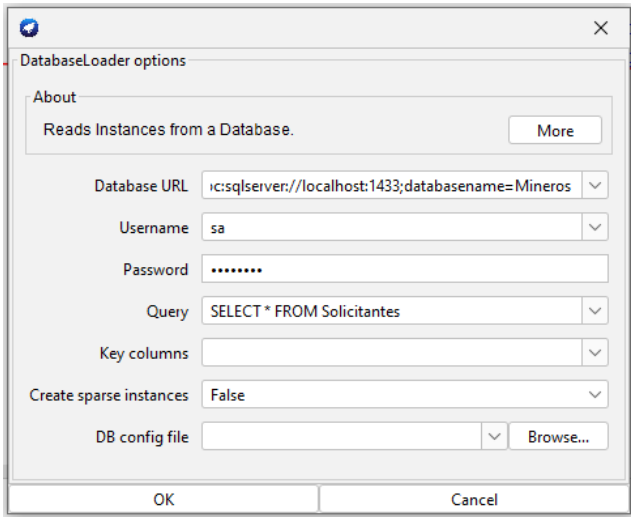




Explicación breve de los componentes

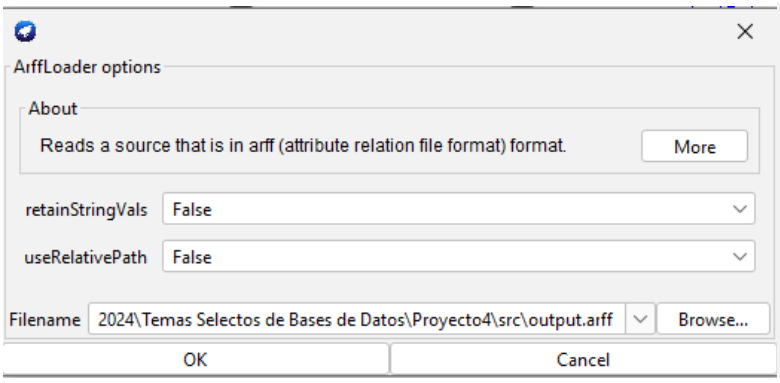
DatabaseLoader

Con este componente, el flujo se puede alimentar de información en una base de datos, en nuestro caso, de la tabla Solicitantes. Cabe decir que la información no necesariamente tiene que pertenecer a una sola tabla, pues, como se aprecia, podemos realizar una consulta tan compleja como queramos.



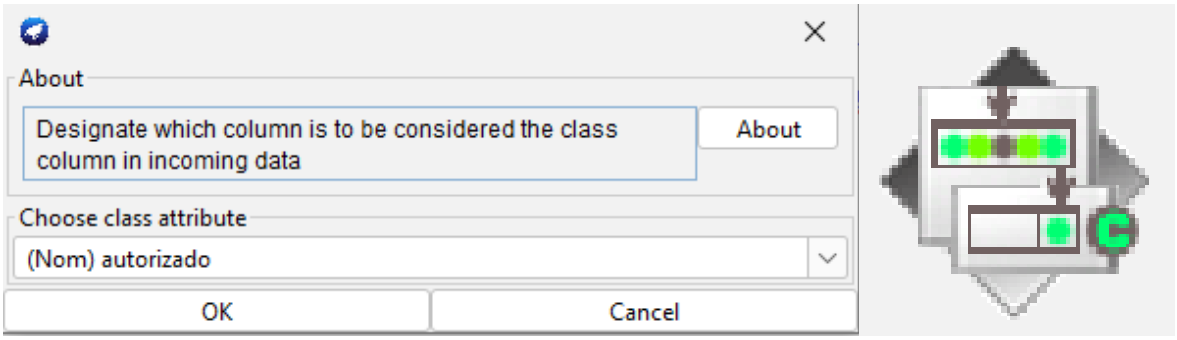
ARFFLoader

Al igual que el anterior, este nos sirve para asignar la fuente de nuestros ejemplos, en este caso, utilizando un archivo ARFF, basta con seleccionar la ruta del archivo y ya está listo (además de asignar las aristas en el flujo).



ClassAssigner

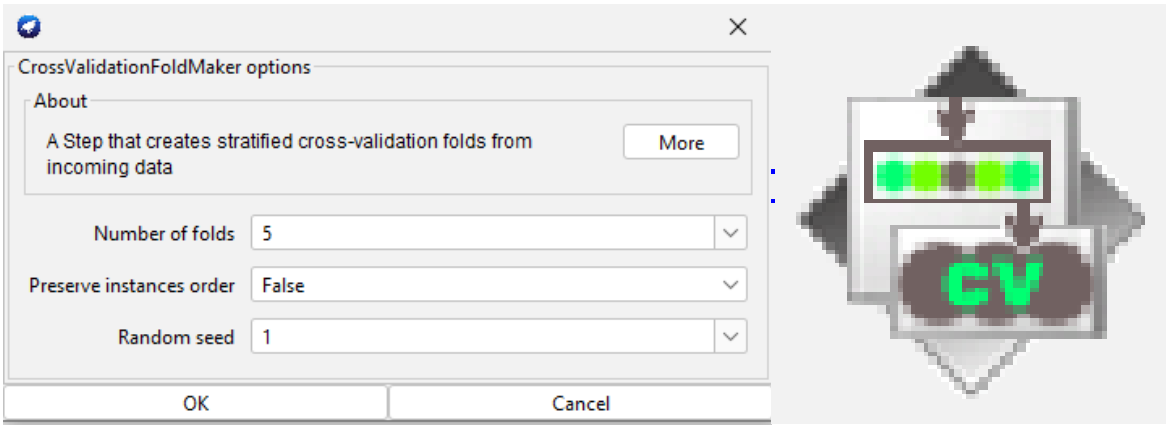
Con este componente se selecciona el atributo por el cual se clasificará posteriormente. Es muy fácil de configurar:



CrossValidationFoldMarker

Con este componente se terminan de establecer algunos criterios, como el número de patrones que queremos generar, en este caso, se establecieron 5.

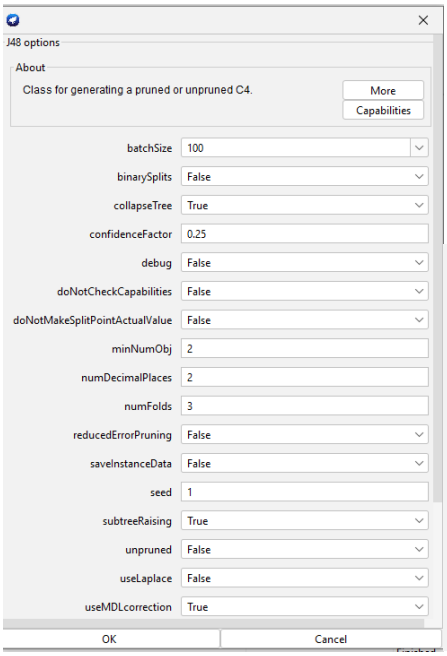
Utiliza un método de validación cruzada, que ayuda a garantizar la efectividad del aprendizaje automático.



Algoritmo de árbol de decision J48 o C4.5

Es el componente estrella del flujo. Este algoritmo construye árboles de decisión basándose en un conjunto de datos de entrenamiento. Este algoritmo es recursivo, intentando cada vez separar por el atributo que tenga la mayor información, o visto de otro modo, que separa de forma más efectiva los datos, basándose en ciertas heurísticas como la diferencia en entropía. De esta manera, el algoritmo produce distintos árboles de decisión que pueden ser útiles para la clasificación de algo, pues tiene que ser validado por un experto.

Su configuración se mantuvo como por defecto:



TextViewer

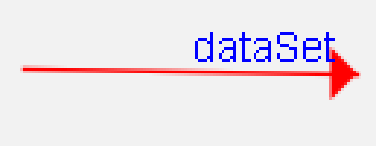
Nos permite visualizar, de manera textual, los patrones generados por el algoritmo, con un formato similar a como denoté las reglas de la generación de datos.

GraphViewer

Nos permite visualizar, de manera gráfica, los patrones generados por el algoritmo, siendo un árbol enraizado, en donde las hojas son valores del atributo elegido para clasificar y en cada nodo se tienen distintas aristas, de las cuales se segmenta la información con respecto a un atributo.

Aristas

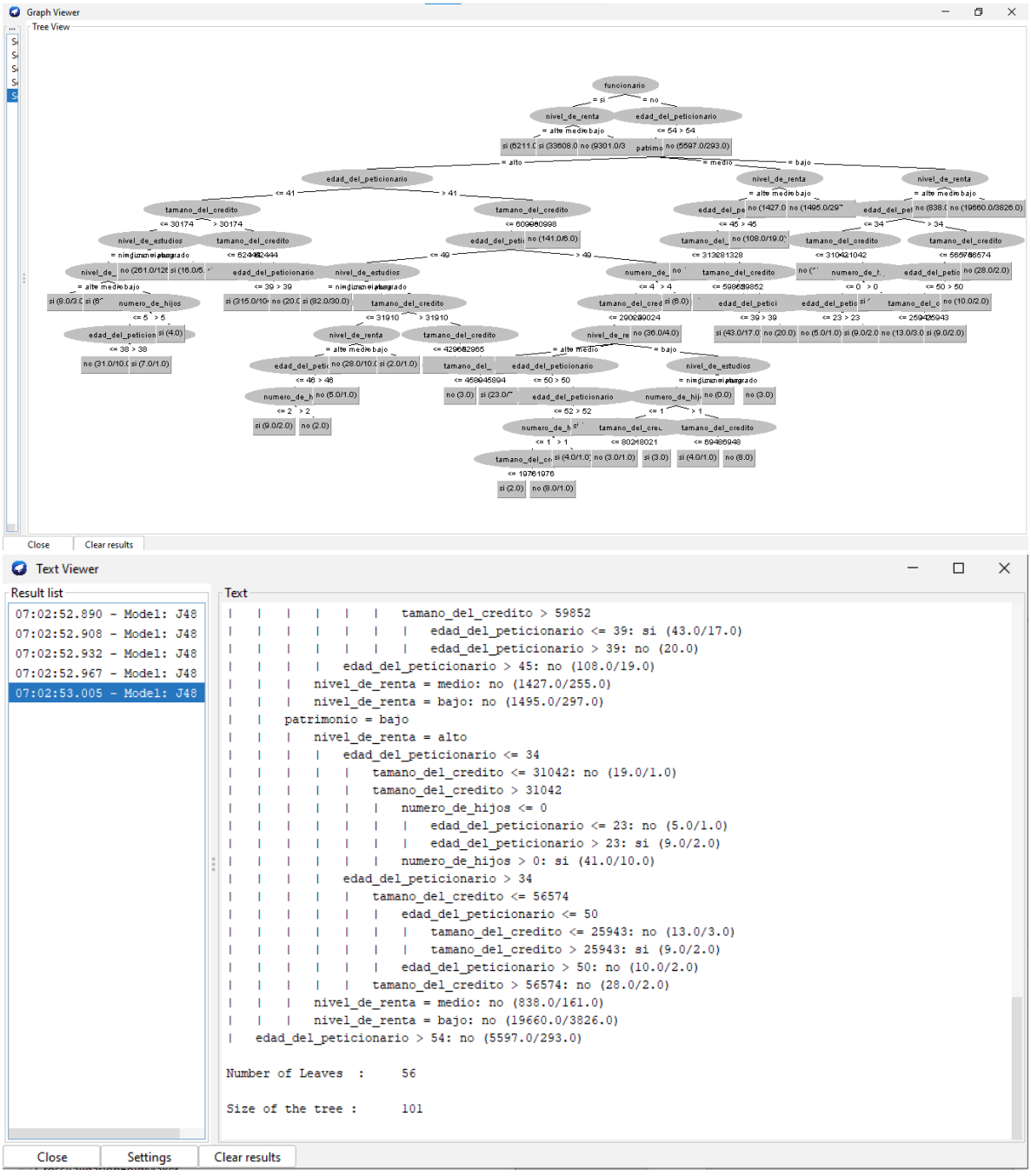
Es la información que se transmite de un componente a otro, en el flujo, los componentes procesan los ejemplos y luego lo pasan al siguiente en el orden.

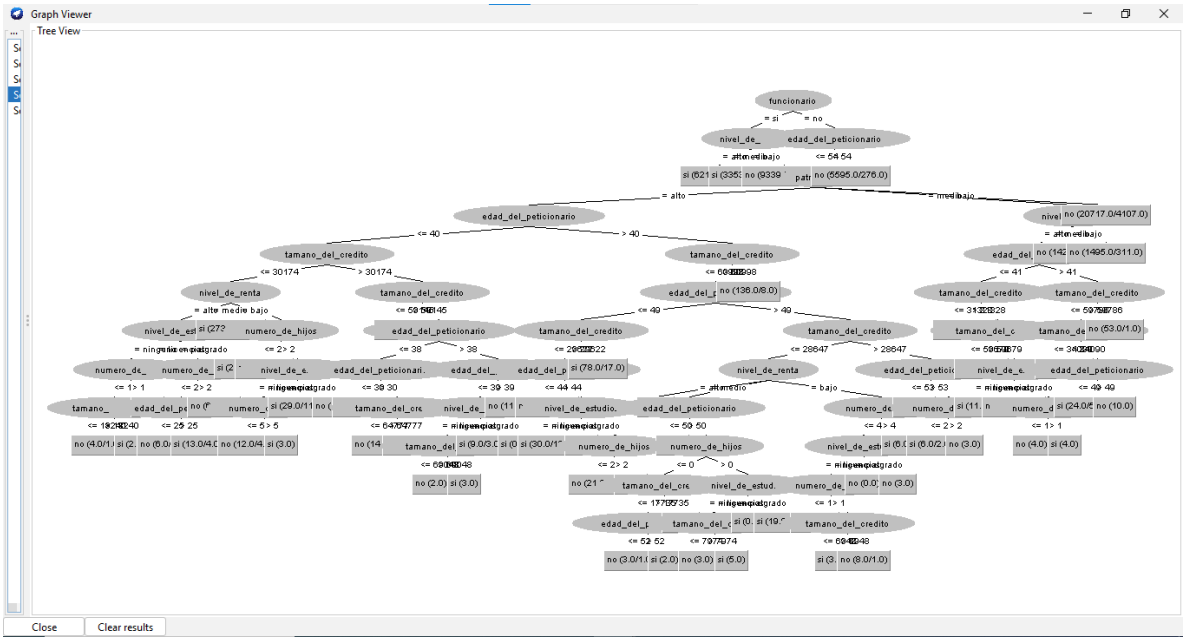


Patrones generados

Los patrones son muy grandes y no logran ajustarse del todo a la pantalla

Criterio. Autorizado





Text Viewer

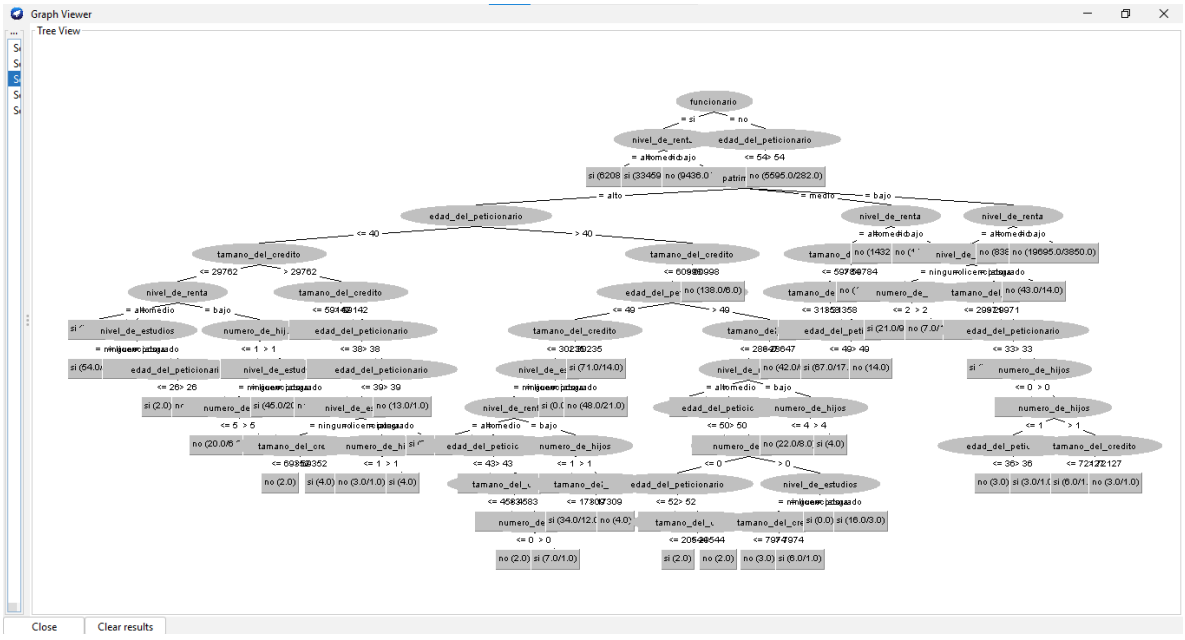
Result list

07:02:52.890 - Model: J48
07:02:52.908 - Model: J48
07:02:52.932 - Model: J48
07:02:52.967 - Model: J48
07:02:53.005 - Model: J48

Text

```
|   |   |   |   |   |   |   numero_de_hijos > 2: no (3.0)
|   |   |   |   tamano_del_credito > 60998: no (136.0/8.0)
|   |   patrimonio = medio
|   |   |   nivel_de_renta = alto
|   |   |   |   edad_del_peticionario <= 41
|   |   |   |   |   tamano_del_credito <= 31328: no (38.0/12.0)
|   |   |   |   |   tamano_del_credito > 31328
|   |   |   |   |   |   tamano_del_credito <= 59679: si (49.0/9.0)
|   |   |   |   |   |   tamano_del_credito > 59679
|   |   |   |   |   |   |   nivel_de_estudios = ninguno: si (11.0/3.0)
|   |   |   |   |   |   |   nivel_de_estudios = licenciatura: no (33.0/15.0)
|   |   |   |   |   |   |   nivel_de_estudios = posgrado
|   |   |   |   |   |   |   |   numero_de_hijos <= 1: no (4.0)
|   |   |   |   |   |   |   |   numero_de_hijos > 1: si (4.0)
|   |   |   |   edad_del_peticionario > 41
|   |   |   |   |   tamano_del_credito <= 59786
|   |   |   |   |   |   tamano_del_credito <= 34090: no (43.0/8.0)
|   |   |   |   |   |   tamano_del_credito > 34090
|   |   |   |   |   |   |   edad_del_peticionario <= 49: si (24.0/5.0)
|   |   |   |   |   |   |   edad_del_peticionario > 49: no (10.0)
|   |   |   |   |   tamano_del_credito > 59786: no (53.0/1.0)
|   |   |   nivel_de_renta = medio: no (1424.0/256.0)
|   |   |   nivel_de_renta = bajo: no (1495.0/311.0)
|   |   patrimonio = bajo: no (20717.0/4107.0)
|   edad_del_peticionario > 54: no (5595.0/276.0)

Number of Leaves  :     64

Size of the tree :      115
```

Close | Settings | Clear results



Text Viewer

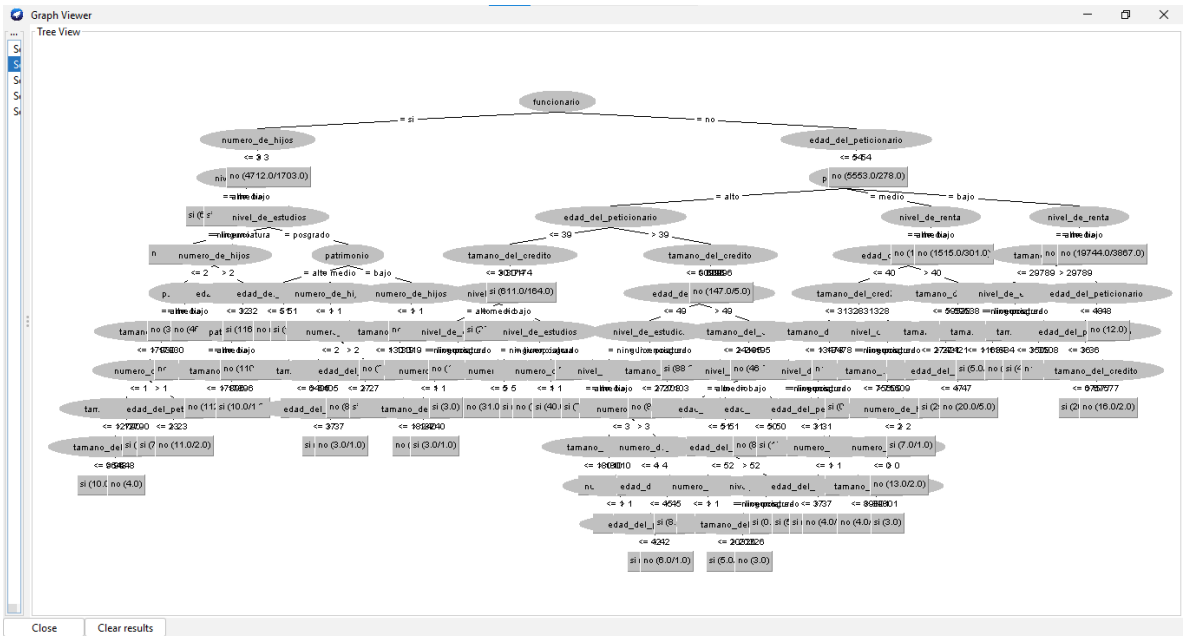
Result list

07:02:52.890 - Model: J48
07:02:52.908 - Model: J48
07:02:52.932 - Model: J48
07:02:52.967 - Model: J48
07:02:53.005 - Model: J48

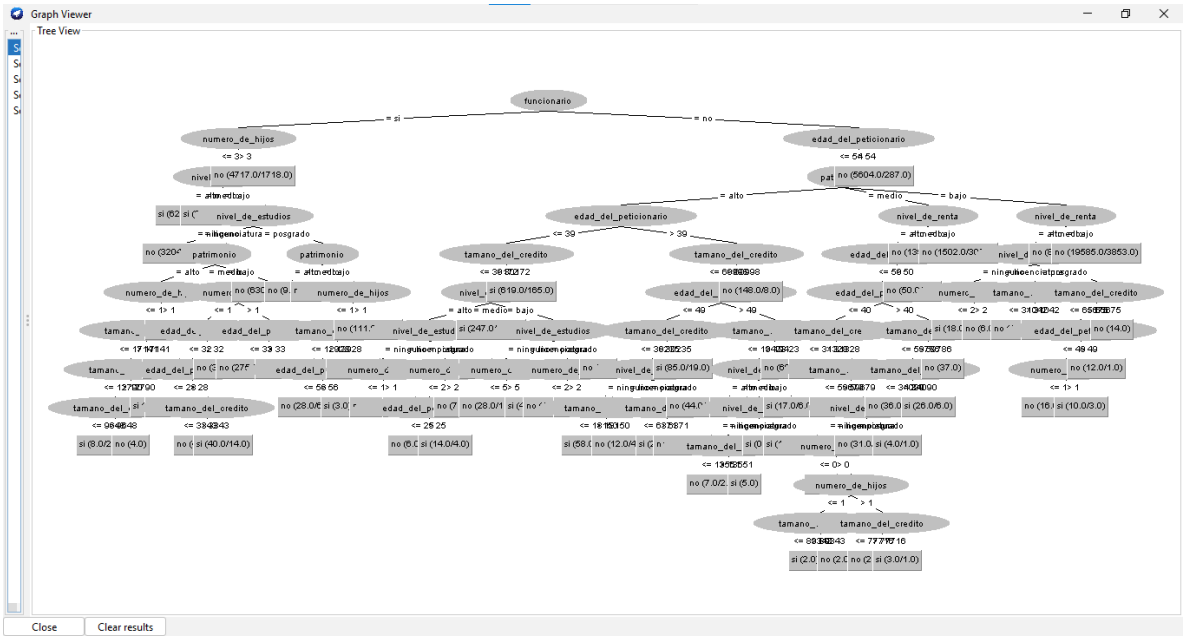
Text

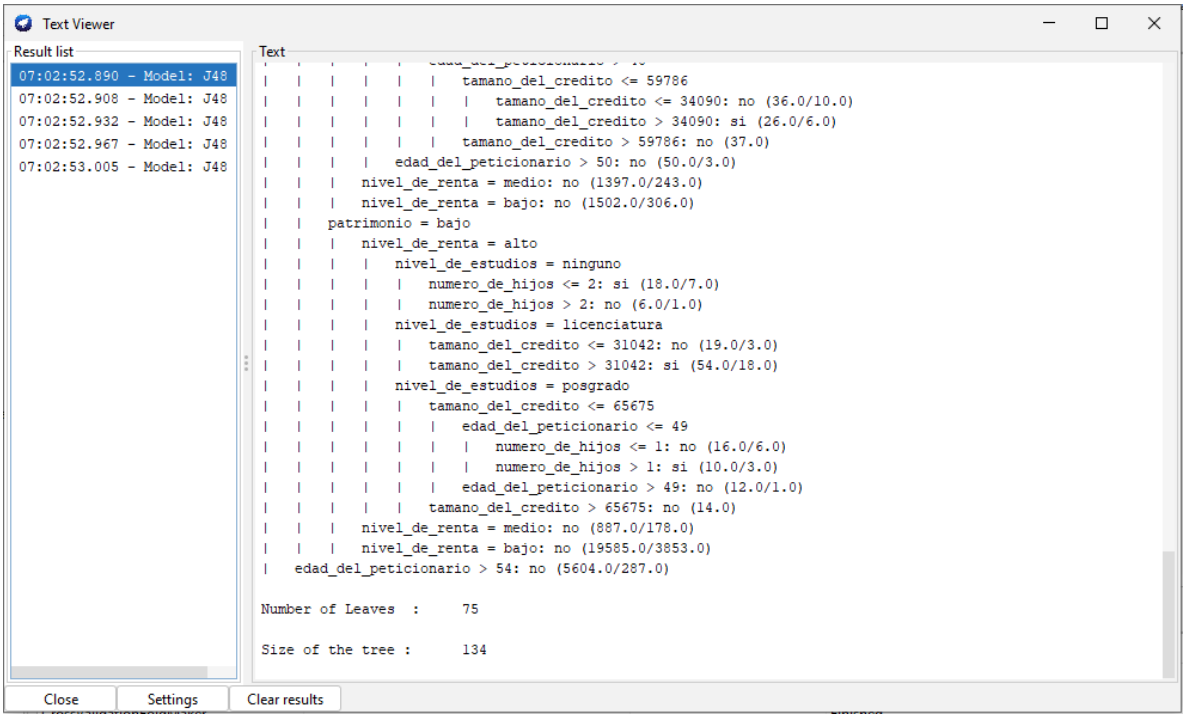
```
|   |   |   |   tamano_del_credito > 59784: no (117.0/30.0)
|   |   |   nivel_de_renta = medio: no (1432.0/270.0)
|   |   |   nivel_de_renta = bajo: no (1481.0/313.0)
|   |   patrimonio = bajo
|   |   |   nivel_de_renta = alto
|   |   |   |   nivel_de_estudios = ninguno
|   |   |   |   |   numero_de_hijos <= 2: si (21.0/9.0)
|   |   |   |   |   numero_de_hijos > 2: no (7.0/1.0)
|   |   |   nivel_de_estudios = licenciatura
|   |   |   |   tamano_del_credito <= 29971: no (15.0/2.0)
|   |   |   |   tamano_del_credito > 29971
|   |   |   |   |   edad_del_peticionario <= 33: si (34.0/7.0)
|   |   |   |   |   edad_del_peticionario > 33
|   |   |   |   |   |   numero_de_hijos <= 0: no (4.0)
|   |   |   |   |   |   numero_de_hijos > 0
|   |   |   |   |   |   |   numero_de_hijos <= 1
|   |   |   |   |   |   |   |   edad_del_peticionario <= 36: no (3.0)
|   |   |   |   |   |   |   |   edad_del_peticionario > 36: si (3.0/1.0)
|   |   |   |   |   |   numero_de_hijos > 1
|   |   |   |   |   |   |   tamano_del_credito <= 72127: si (6.0/1.0)
|   |   |   |   |   |   |   tamano_del_credito > 72127: no (3.0/1.0)
|   |   |   |   nivel_de_estudios = posgrado: no (43.0/14.0)
|   |   |   nivel_de_renta = medio: no (838.0/161.0)
|   |   |   nivel_de_renta = bajo: no (19695.0/3850.0)
|   edad_del_peticionario > 54: no (5595.0/282.0)

Number of Leaves  :     64

Size of the tree :      114
```

Close | Settings | Clear results

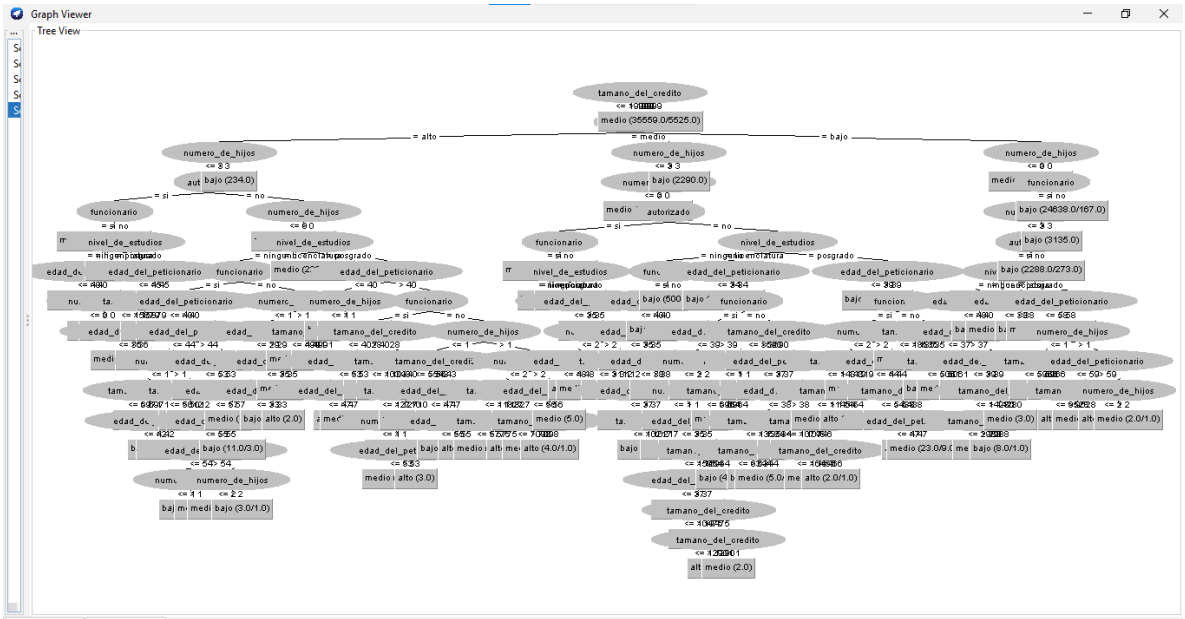

Graph Viewer
Tree View
funcionario
= si
= no
numero_de_hijos
edad_del_peticionario
<= 54
no (5553.0/278.0)
= alto
= medio
= bajo
edad_del_peticionario
nivel_de_renta
nivel_de_renta
nivel_de_estudios
= ninguno
= posgrado
numero_de_hijos
patrimonio
tamano_del_credito
tamano_del_credito
<= 39
> 39
<= 2
> 2
= alto
= medio
= bajo
si (611.0/164.0)
<= 49
> 49
<= 40
> 40
<= 29789
> 29789
no (19744.0/3867.0)
Close
Clear results

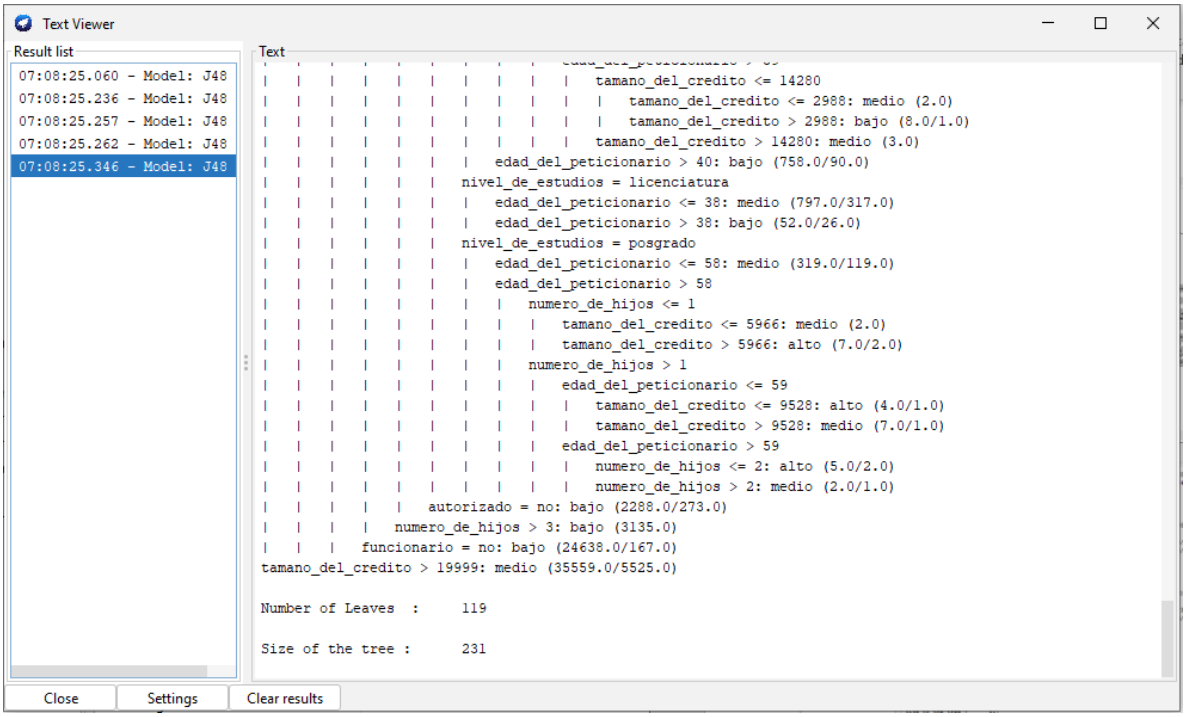

Text Viewer
Result list
07:02:52.890 - Model: J48
07:02:52.908 - Model: J48
07:02:52.932 - Model: J48
07:02:52.967 - Model: J48
07:02:53.005 - Model: J48
Text
| | | | | | | edad_del_peticionario <= 47: si (25.0/6.0)
| | | | | | | edad_del_peticionario > 47: no (20.0/5.0)
| | | | | tamano_del_credito > 59538: no (56.0/2.0)
| | | nivel_de_renta = medio: no (1380.0/256.0)
| | | nivel_de_renta = bajo: no (1515.0/301.0)
| | patrimonio = bajo
| | | nivel_de_renta = alto
| | | | tamano_del_credito <= 29789
| | | | | nivel_de_estudios = ninguno
| | | | | | tamano_del_credito <= 11634: si (5.0/1.0)
| | | | | | tamano_del_credito > 11634: no (4.0)
| | | | | nivel_de_estudios = licenciatura: no (22.0/3.0)
| | | | | nivel_de_estudios = posgrado
| | | | | | tamano_del_credito <= 3508: si (4.0/1.0)
| | | | | | tamano_del_credito > 3508: no (12.0/1.0)
| | | | tamano_del_credito > 29789
| | | | | edad_del_peticionario <= 48
| | | | | | edad_del_peticionario <= 36: si (63.0/18.0)
| | | | | | edad_del_peticionario > 36
| | | | | | | tamano_del_credito <= 67577: si (20.0/5.0)
| | | | | | | tamano_del_credito > 67577: no (16.0/2.0)
| | | | | edad_del_peticionario > 48: no (12.0)
| | | nivel_de_renta = medio: no (856.0/161.0)
| | | nivel_de_renta = bajo: no (19744.0/3867.0)
| edad_del_peticionario > 54: no (5553.0/278.0)
Number of Leaves : 96
Size of the tree : 173
Close
Settings
Clear results

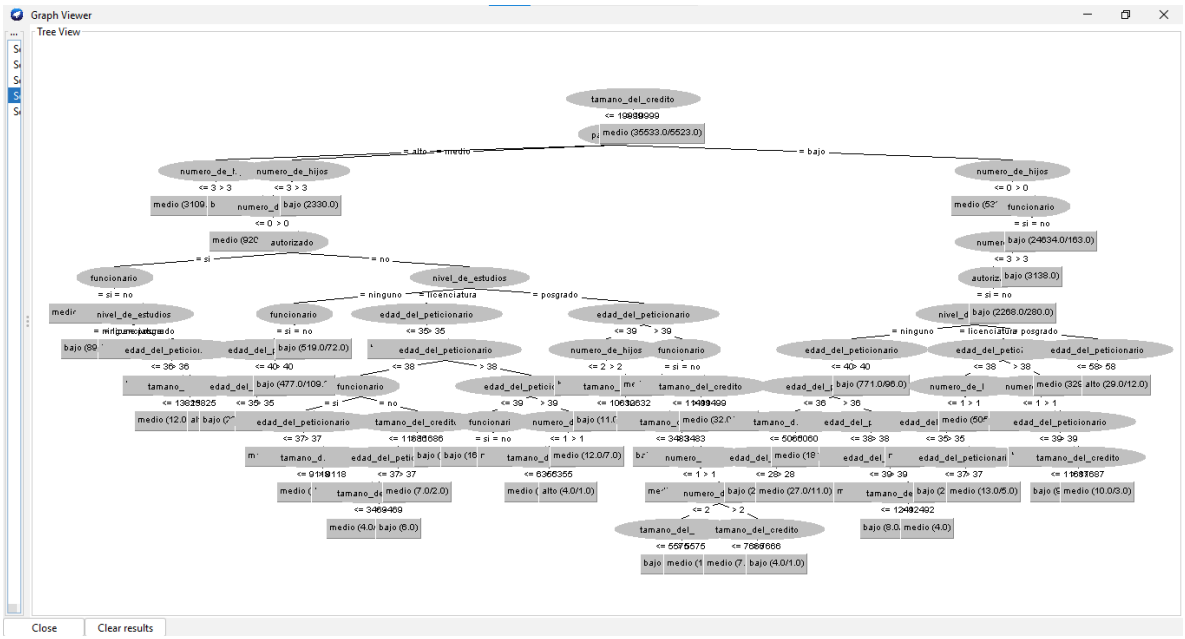

Graph Viewer
Tree View
funcionario
= si
= no
numero_de_hijos
<= 3
> 3
edad_del_peticionario
<= 54
> 54
= alto
= medio
= bajo
nivel_de_renta
nivel_de_estudios
patrimonio
tamano_del_credito
Close
Clear results

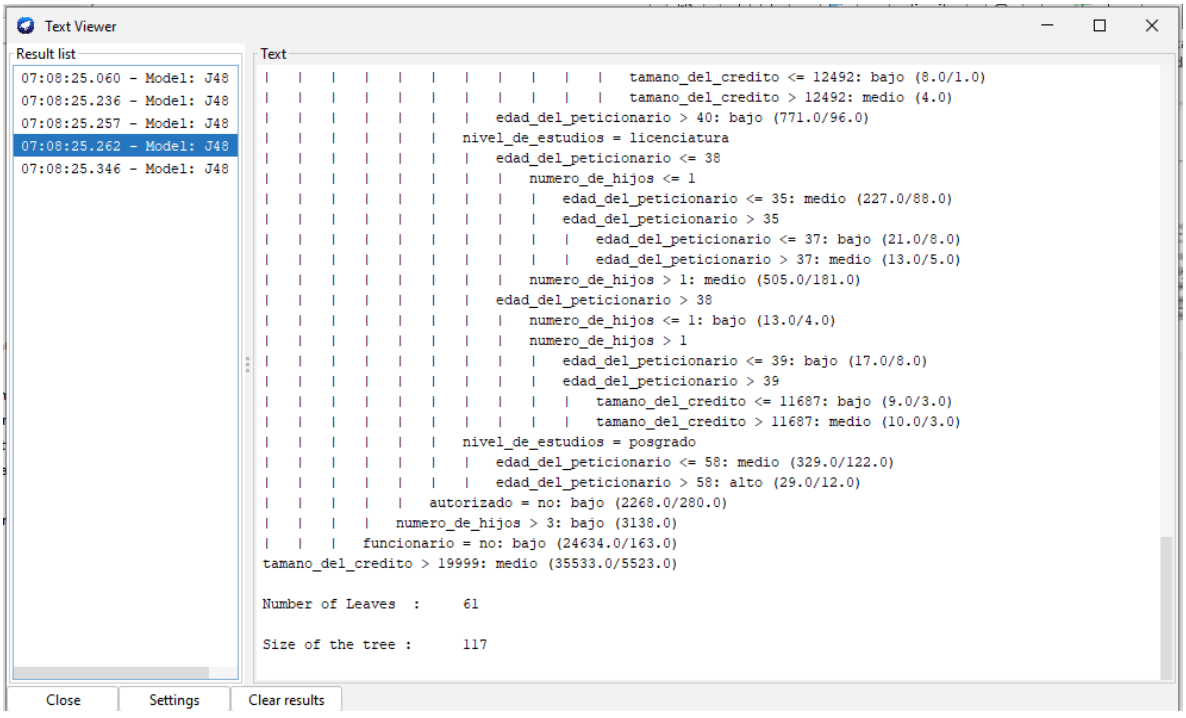

Text Viewer
Result list
07:02:52.890 - Model: J48
07:02:52.908 - Model: J48
07:02:52.932 - Model: J48
07:02:52.967 - Model: J48
07:02:53.005 - Model: J48
Text
| | | | | | | tamano_del_credito <= 59786
| | | | | | | | tamano_del_credito <= 34090: no (36.0/10.0)
| | | | | | | | tamano_del_credito > 34090: si (26.0/6.0)
| | | | | | | tamano_del_credito > 59786: no (37.0)
| | | | | edad_del_peticionario > 50: no (50.0/3.0)
| | | nivel_de_renta = medio: no (1397.0/243.0)
| | | nivel_de_renta = bajo: no (1502.0/306.0)
| | patrimonio = bajo
| | | nivel_de_renta = alto
| | | | nivel_de_estudios = ninguno
| | | | | numero_de_hijos <= 2: si (18.0/7.0)
| | | | | numero_de_hijos > 2: no (6.0/1.0)
| | | | nivel_de_estudios = licenciatura
| | | | | tamano_del_credito <= 31042: no (19.0/3.0)
| | | | | tamano_del_credito > 31042: si (54.0/18.0)
| | | | nivel_de_estudios = posgrado
| | | | | tamano_del_credito <= 65675
| | | | | | edad_del_peticionario <= 49
| | | | | | | numero_de_hijos <= 1: no (16.0/6.0)
| | | | | | | numero_de_hijos > 1: si (10.0/3.0)
| | | | | | edad_del_peticionario > 49: no (12.0/1.0)
| | | | | tamano_del_credito > 65675: no (14.0)
| | | nivel_de_renta = medio: no (887.0/178.0)
| | | nivel_de_renta = bajo: no (19585.0/3853.0)
| edad_del_peticionario > 54: no (5604.0/287.0)
Number of Leaves : 75
Size of the tree : 134
Close
Settings
Clear results

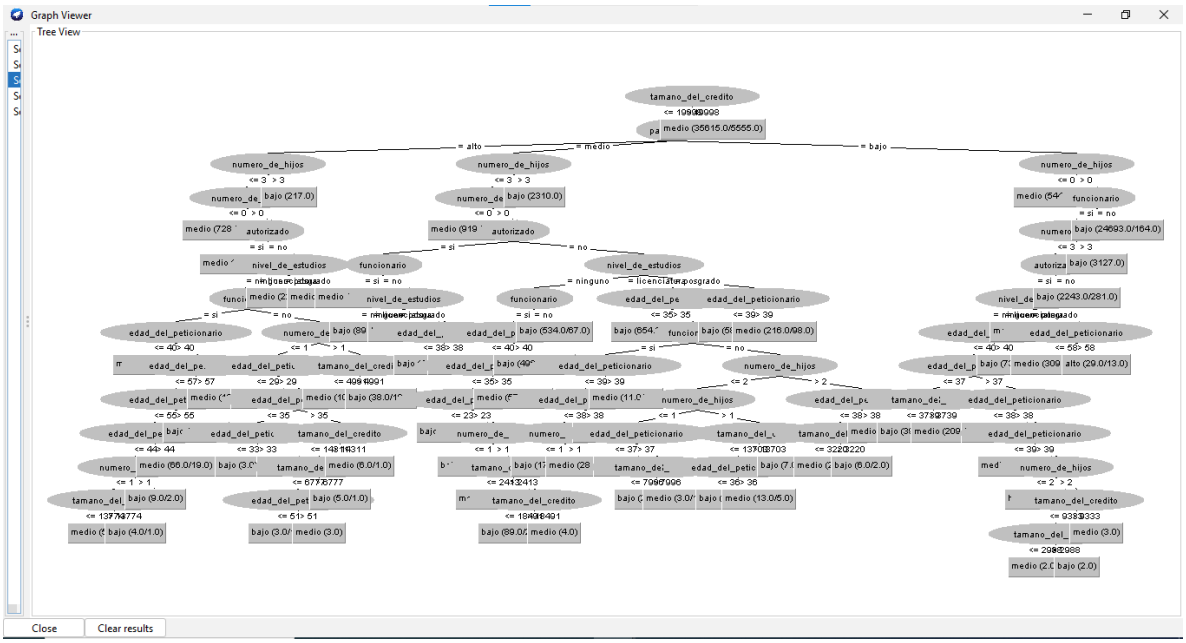
Criterio. Nivel de renta






Graph Viewer
Tree View
tamano_del_credito
<= 19999999
medio (35533.0/5523.0)
= alto = medio
= bajo
numero_de_hijos
<= 0 > 0
= si = no
bajo (24634.0/163.0)
<= 3 > 3
bajo (3138.0)
= si = no
bajo (2268.0/280.0)
= ninguno = licenciatura = posgrado
edad_del_peticionario
<= 40 > 40
<= 38 > 38
<= 58 > 58
bajo (771.0/96.0)
medio (329.0/122.0)
alto (29.0/12.0)
<= 1 > 1
medio (227.0/88.0)
edad_del_peticionario
<= 37 > 37
<= 35 > 35
bajo (21.0/8.0)
medio (13.0/5.0)
numero_de_hijos
<= 1 > 1
medio (505.0/181.0)
<= 2 > 2
bajo (13.0/4.0)
medio (27.0/11.0)
<= 39 > 39
bajo (17.0/8.0)
medio (10.0/3.0)
tamano_del_credito
<= 11687 > 11687
bajo (9.0/3.0)
medio (4.0)
Close
Clear results


Text Viewer
Result list
07:08:25.060 - Model: J48
07:08:25.236 - Model: J48
07:08:25.257 - Model: J48
07:08:25.262 - Model: J48
07:08:25.346 - Model: J48
Text
| | | | | | | | | | | | tamano_del_credito <= 12492: bajo (8.0/1.0)
| | | | | | | | | | | | tamano_del_credito > 12492: medio (4.0)
| | | | | | | | edad_del_peticionario > 40: bajo (771.0/96.0)
| | | | | | | nivel_de_estudios = licenciatura
| | | | | | | | edad_del_peticionario <= 38
| | | | | | | | | numero_de_hijos <= 1
| | | | | | | | | | edad_del_peticionario <= 35: medio (227.0/88.0)
| | | | | | | | | | edad_del_peticionario > 35
| | | | | | | | | | | edad_del_peticionario <= 37: bajo (21.0/8.0)
| | | | | | | | | | | edad_del_peticionario > 37: medio (13.0/5.0)
| | | | | | | | | numero_de_hijos > 1: medio (505.0/181.0)
| | | | | | | | edad_del_peticionario > 38
| | | | | | | | | numero_de_hijos <= 1: bajo (13.0/4.0)
| | | | | | | | | numero_de_hijos > 1
| | | | | | | | | | edad_del_peticionario <= 39: bajo (17.0/8.0)
| | | | | | | | | | edad_del_peticionario > 39
| | | | | | | | | | | tamano_del_credito <= 11687: bajo (9.0/3.0)
| | | | | | | | | | | tamano_del_credito > 11687: medio (10.0/3.0)
| | | | | | | nivel_de_estudios = posgrado
| | | | | | | | edad_del_peticionario <= 58: medio (329.0/122.0)
| | | | | | | | edad_del_peticionario > 58: alto (29.0/12.0)
| | | | | | | autorizado = no: bajo (2268.0/280.0)
| | | | | numero_de_hijos > 3: bajo (3138.0)
| | | | funcionario = no: bajo (24634.0/163.0)
tamano_del_credito > 19999: medio (35533.0/5523.0)
Number of Leaves : 61
Size of the tree : 117
Close
Settings
Clear results



Text Viewer

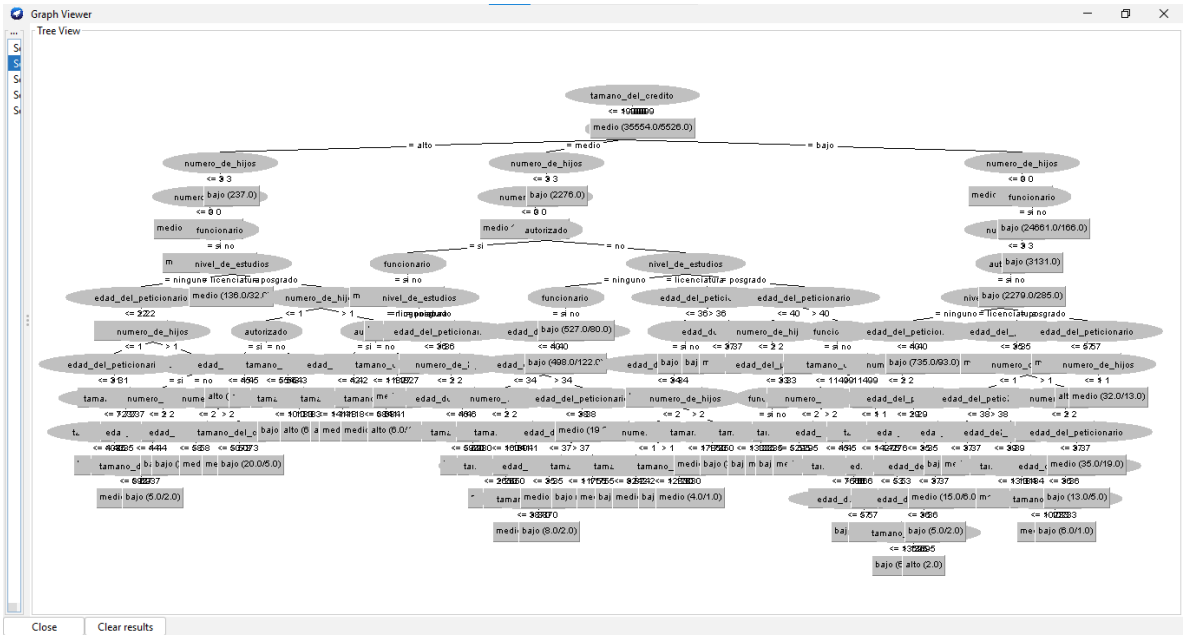
Result list

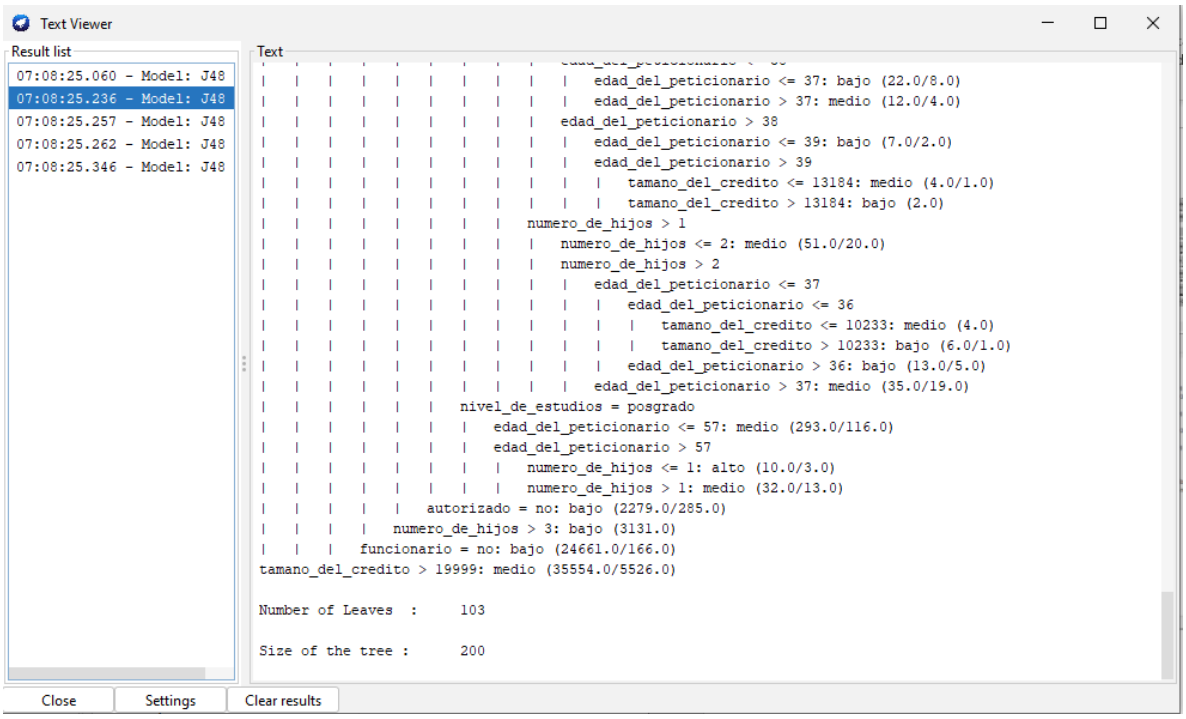
07:08:25.060 - Model: J48
07:08:25.236 - Model: J48
07:08:25.257 - Model: J48
07:08:25.262 - Model: J48
07:08:25.346 - Model: J48

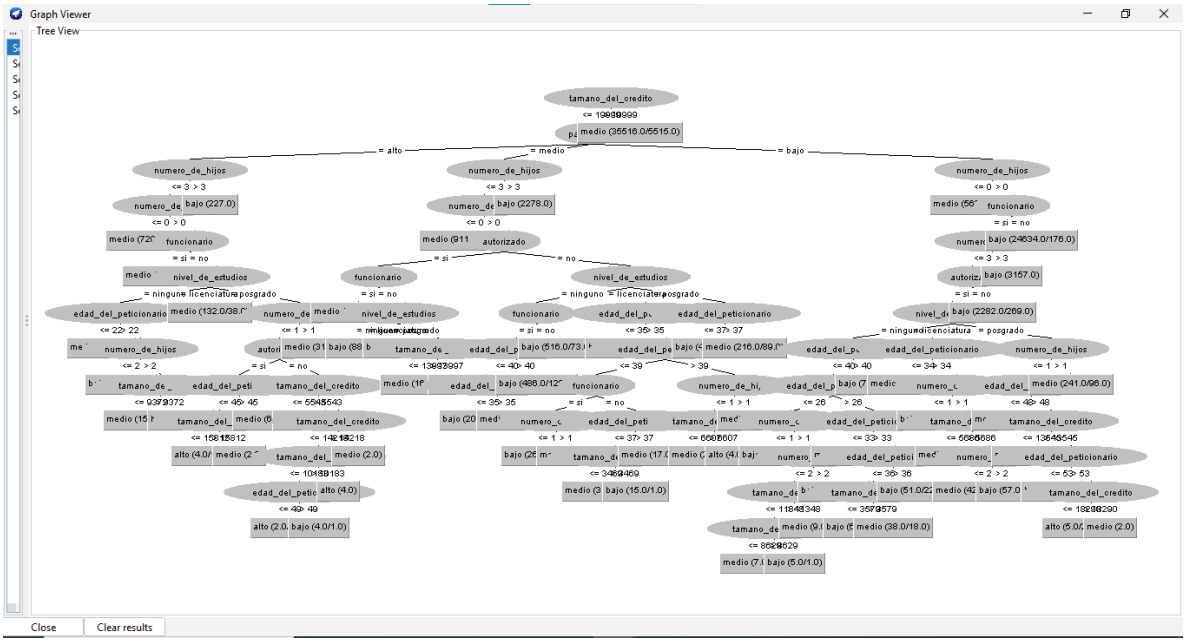
Text

```
|   |   |   |   |   |   nivel_de_estudios = ninguno
|   |   |   |   |   |   |   edad_del_peticionario <= 40
|   |   |   |   |   |   |   |   edad_del_peticionario <= 37
|   |   |   |   |   |   |   |   |   tamano_del_credito <= 3739: bajo (39.0/13.0)
|   |   |   |   |   |   |   |   |   tamano_del_credito > 3739: medio (209.0/97.0)
|   |   |   |   |   |   |   |   edad_del_peticionario > 37
|   |   |   |   |   |   |   |   |   edad_del_peticionario <= 38: bajo (13.0/4.0)
|   |   |   |   |   |   |   |   |   edad_del_peticionario > 38
|   |   |   |   |   |   |   |   |   |   edad_del_peticionario <= 39: medio (16.0/5.0)
|   |   |   |   |   |   |   |   |   |   edad_del_peticionario > 39
|   |   |   |   |   |   |   |   |   |   |   numero_de_hijos <= 2: bajo (6.0/1.0)
|   |   |   |   |   |   |   |   |   |   |   numero_de_hijos > 2
|   |   |   |   |   |   |   |   |   |   |   |   tamano_del_credito <= 9333
|   |   |   |   |   |   |   |   |   |   |   |   |   tamano_del_credito <= 2988: medio (2.0)
|   |   |   |   |   |   |   |   |   |   |   |   |   tamano_del_credito > 2988: bajo (2.0)
|   |   |   |   |   |   |   |   |   |   |   |   tamano_del_credito > 9333: medio (3.0)
|   |   |   |   |   |   |   edad_del_peticionario > 40: bajo (735.0/89.0)
|   |   |   |   |   |   nivel_de_estudios = licenciatura: medio (828.0/328.0)
|   |   |   |   |   |   nivel_de_estudios = posgrado
|   |   |   |   |   |   |   edad_del_peticionario <= 58: medio (309.0/119.0)
|   |   |   |   |   |   |   edad_del_peticionario > 58: alto (29.0/13.0)
|   |   |   |   |   autorizado = no: bajo (2243.0/281.0)
|   |   |   |   numero_de_hijos > 3: bajo (3127.0)
|   |   |   funcionario = no: bajo (24693.0/164.0)
tamano_del_credito > 19998: medio (35615.0/5555.0)

Number of Leaves  :     69

Size of the tree :      132
```

Close | Settings | Clear results


Graph Viewer
Tree View
tamano_del_credito
medio (35554.0/5526.0)
= alto
= medio
= bajo
Close
Clear results


Text Viewer
Result list
07:08:25.060 - Model: J48
07:08:25.236 - Model: J48
07:08:25.257 - Model: J48
07:08:25.262 - Model: J48
07:08:25.346 - Model: J48
Text
| | | | | | | | | | | edad_del_peticionario <= 37: bajo (22.0/8.0)
| | | | | | | | | | | edad_del_peticionario > 37: medio (12.0/4.0)
| | | | | | | | | | edad_del_peticionario > 38
| | | | | | | | | | | edad_del_peticionario <= 39: bajo (7.0/2.0)
| | | | | | | | | | | edad_del_peticionario > 39
| | | | | | | | | | | | tamano_del_credito <= 13184: medio (4.0/1.0)
| | | | | | | | | | | | tamano_del_credito > 13184: bajo (2.0)
| | | | | | | | | numero_de_hijos > 1
| | | | | | | | | | numero_de_hijos <= 2: medio (51.0/20.0)
| | | | | | | | | | numero_de_hijos > 2
| | | | | | | | | | | edad_del_peticionario <= 37
| | | | | | | | | | | | edad_del_peticionario <= 36
| | | | | | | | | | | | | tamano_del_credito <= 10233: medio (4.0)
| | | | | | | | | | | | | tamano_del_credito > 10233: bajo (6.0/1.0)
| | | | | | | | | | | | edad_del_peticionario > 36: bajo (13.0/5.0)
| | | | | | | | | | | edad_del_peticionario > 37: medio (35.0/19.0)
| | | | | | | | nivel_de_estudios = posgrado
| | | | | | | | | edad_del_peticionario <= 57: medio (293.0/116.0)
| | | | | | | | | edad_del_peticionario > 57
| | | | | | | | | | numero_de_hijos <= 1: alto (10.0/3.0)
| | | | | | | | | | numero_de_hijos > 1: medio (32.0/13.0)
| | | | | | autorizado = no: bajo (2279.0/285.0)
| | | | | numero_de_hijos > 3: bajo (3131.0)
| | | | funcionario = no: bajo (24661.0/166.0)
tamano_del_credito > 19999: medio (35554.0/5526.0)
Number of Leaves : 103
Size of the tree : 200
Close
Settings
Clear results



Text Viewer

Result list

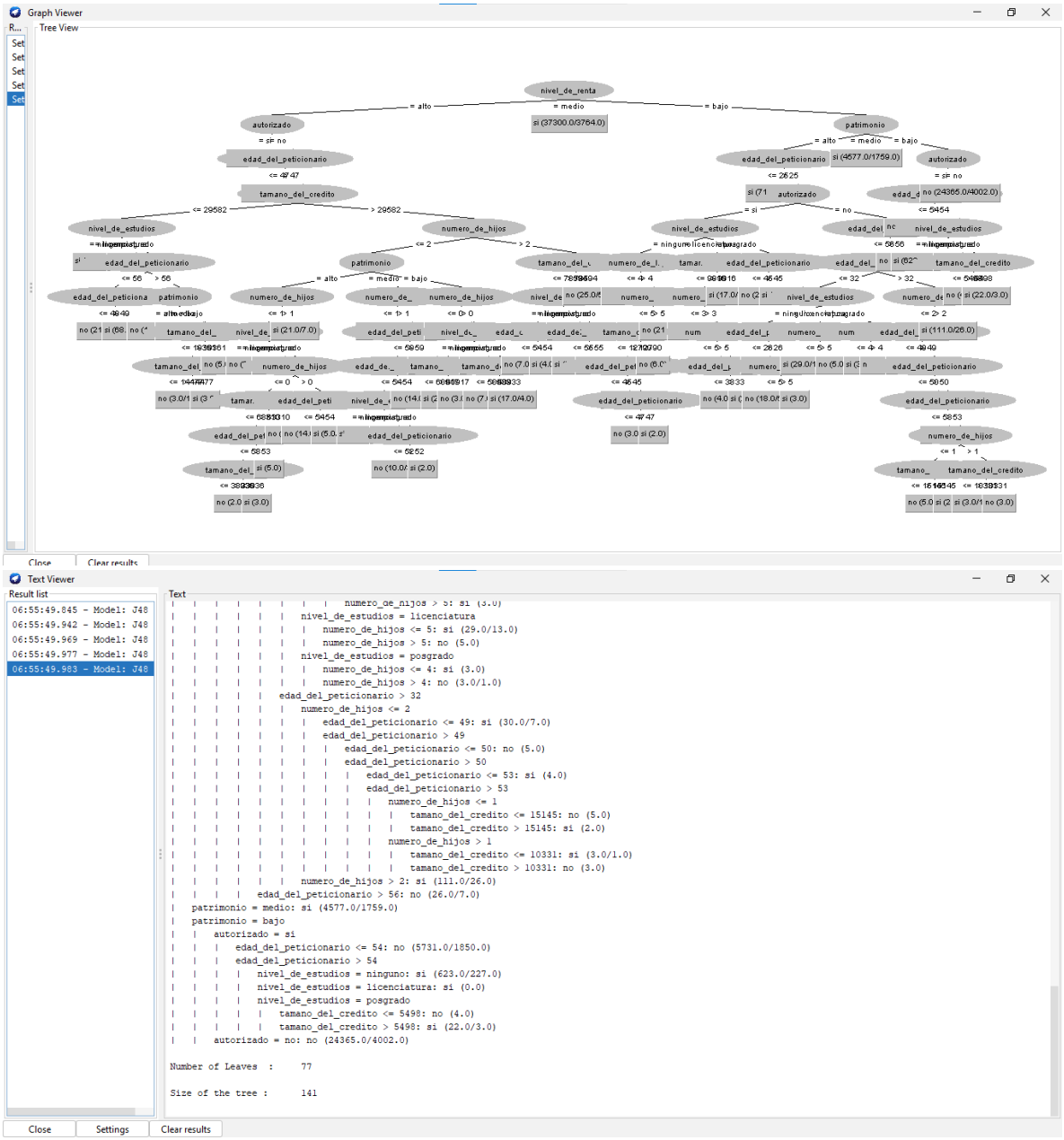
07:08:25.060 - Model: J48
07:08:25.236 - Model: J48
07:08:25.257 - Model: J48
07:08:25.262 - Model: J48
07:08:25.346 - Model: J48

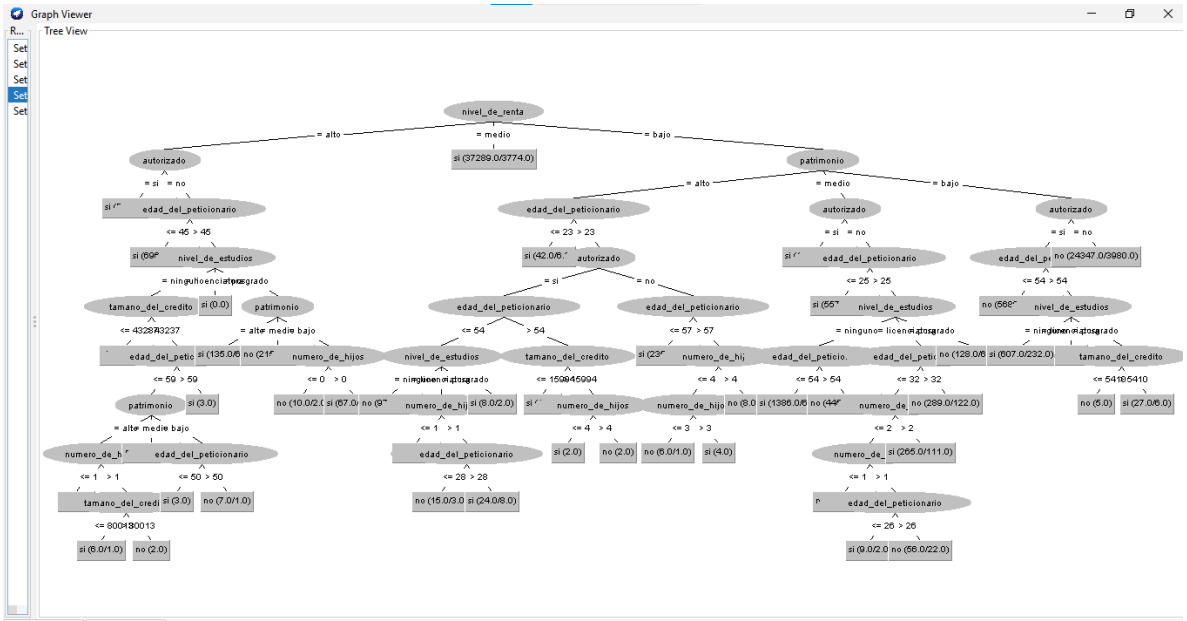
Text

```
|   |   |   |   |   |   |   edad_del_peticionario > 40: bajo (749.0/88.0)
|   |   |   |   |   |   nivel_de_estudios = licenciatura
|   |   |   |   |   |   |   edad_del_peticionario <= 34: medio (655.0/255.0)
|   |   |   |   |   |   |   edad_del_peticionario > 34
|   |   |   |   |   |   |   |   numero_de_hijos <= 1: bajo (67.0/33.0)
|   |   |   |   |   |   |   |   numero_de_hijos > 1
|   |   |   |   |   |   |   |   |   tamano_del_credito <= 5686: medio (37.0/8.0)
|   |   |   |   |   |   |   |   |   tamano_del_credito > 5686
|   |   |   |   |   |   |   |   |   |   numero_de_hijos <= 2: medio (42.0/16.0)
|   |   |   |   |   |   |   |   |   |   numero_de_hijos > 2: bajo (57.0/27.0)
|   |   |   |   |   |   nivel_de_estudios = posgrado
|   |   |   |   |   |   |   numero_de_hijos <= 1
|   |   |   |   |   |   |   |   edad_del_peticionario <= 48: medio (56.0/18.0)
|   |   |   |   |   |   |   |   edad_del_peticionario > 48
|   |   |   |   |   |   |   |   |   tamano_del_credito <= 13545: medio (38.0/15.0)
|   |   |   |   |   |   |   |   |   tamano_del_credito > 13545
|   |   |   |   |   |   |   |   |   |   edad_del_peticionario <= 53: bajo (9.0/2.0)
|   |   |   |   |   |   |   |   |   |   edad_del_peticionario > 53
|   |   |   |   |   |   |   |   |   |   |   tamano_del_credito <= 18290: alto (5.0/2.0)
|   |   |   |   |   |   |   |   |   |   |   tamano_del_credito > 18290: medio (2.0)
|   |   |   |   |   |   |   numero_de_hijos > 1: medio (241.0/96.0)
|   |   |   |   |   |   autorizado = no: bajo (2282.0/269.0)
|   |   |   |   |   numero_de_hijos > 3: bajo (3157.0)
|   |   |   |   funcionario = no: bajo (24634.0/176.0)
tamano_del_credito > 19999: medio (35516.0/5515.0)

Number of Leaves  :     65

Size of the tree :      124
```

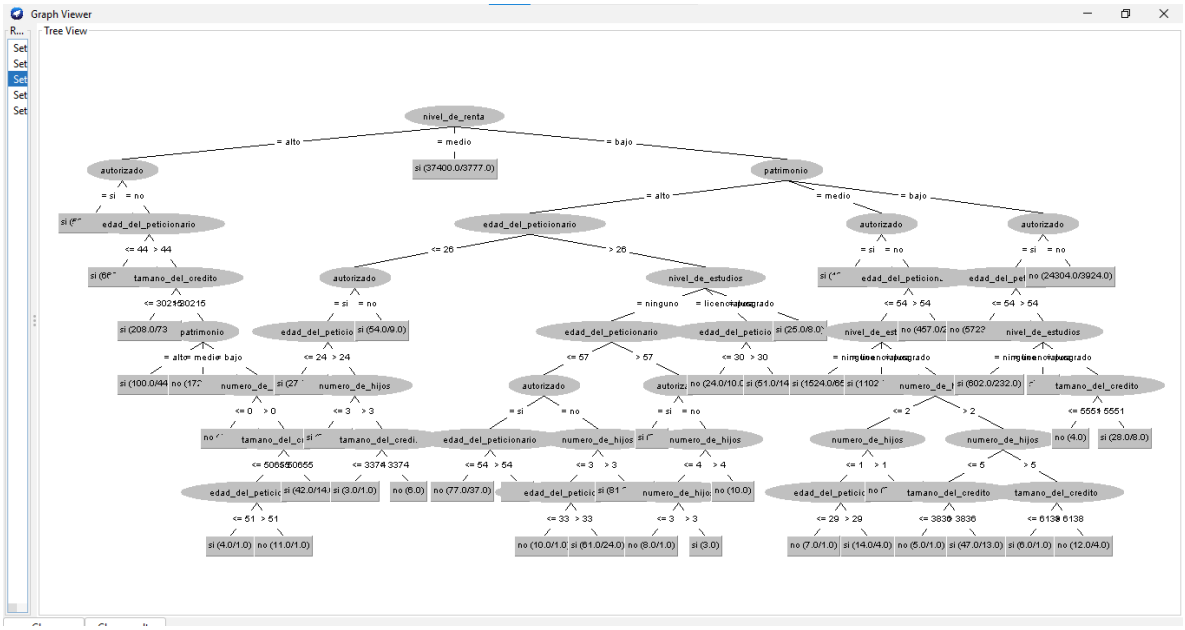
Close | Settings | Clear results

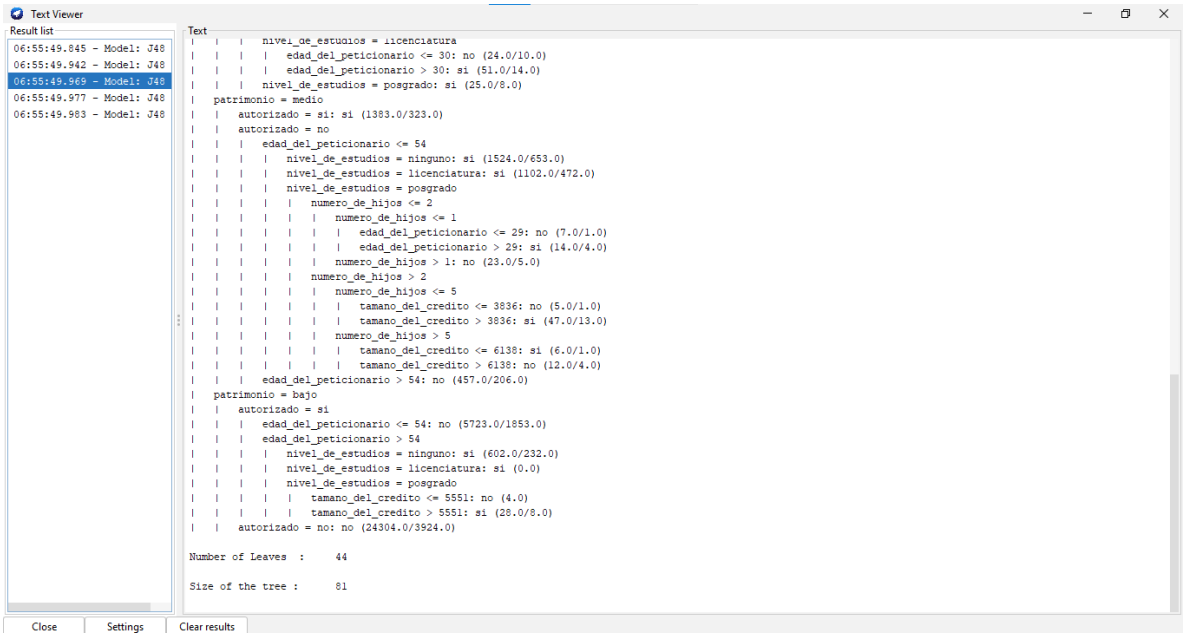
Criterio. Funcionario

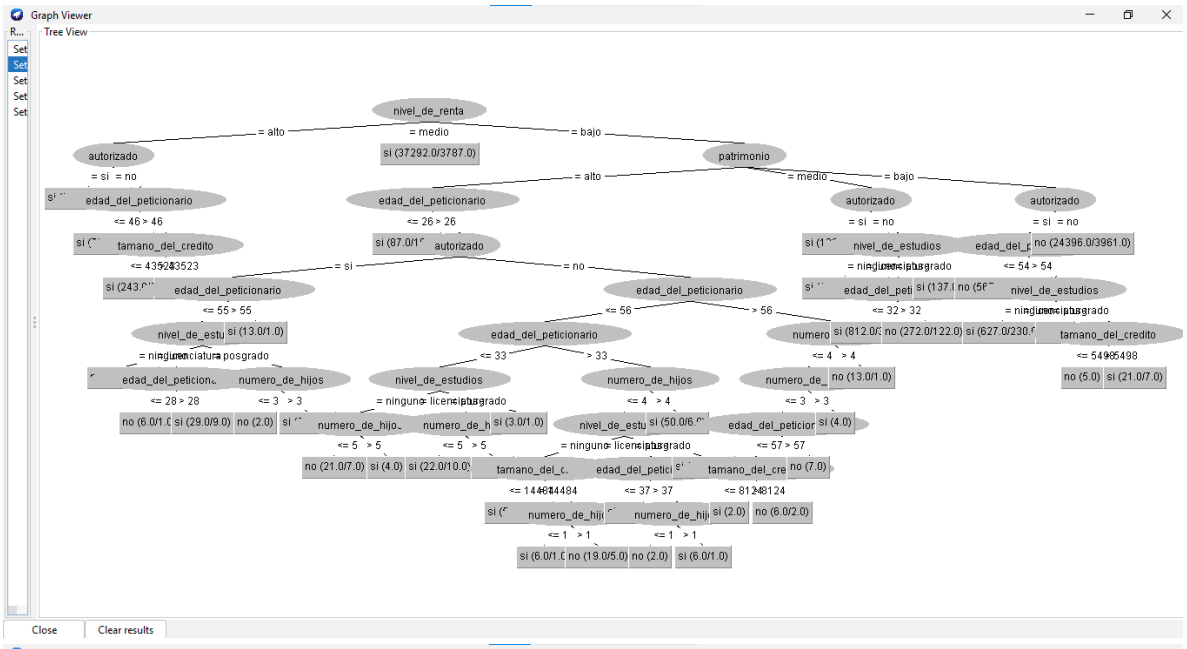


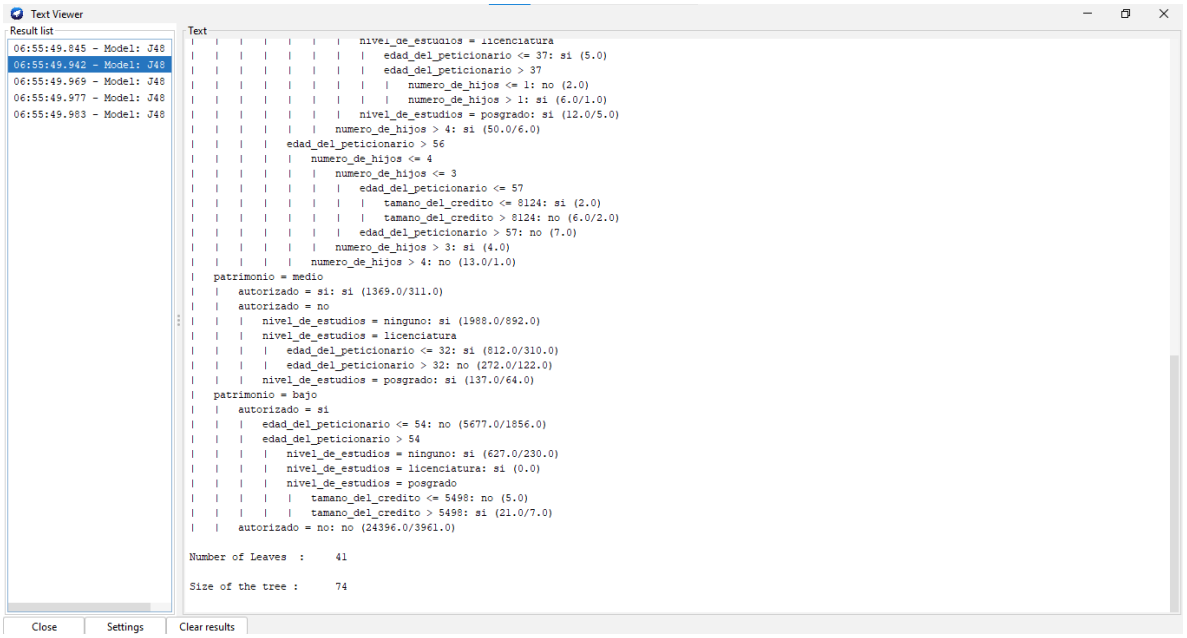

Graph Viewer
Tree View
nivel_de_renta
= alto
= medio
= bajo
autorizado
si (37289.0/3774.0)
patrimonio
= si
= no
edad_del_peticionario
<= 45
> 45
nivel_de_estudios
= ninguno
licenciatura
posgrado
tamano_del_credito
si (3.0)
patrimonio
<= 59
> 59
= alto
medio
bajo
numero_de_hijos
<= 0
> 0
<= 1
> 1
<= 50
> 50
si (3.0)
no (7.0/1.0)
<= 80013
si (6.0/1.0)
no (2.0)
<= 54
> 54
<= 28
> 28
no (15.0/3.0)
si (24.0/8.0)
<= 57
> 57
<= 4
> 4
<= 3
> 3
si (2.0)
no (2.0)
no (6.0/1.0)
si (4.0)
<= 25
> 25
<= 2
> 2
si (265.0/111.0)
<= 26
> 26
si (9.0/2.0)
no (56.0/22.0)
<= 54
> 54
no (24347.0/3980.0)
si (27.0/6.0)
no (5.0)
Close
Clear results

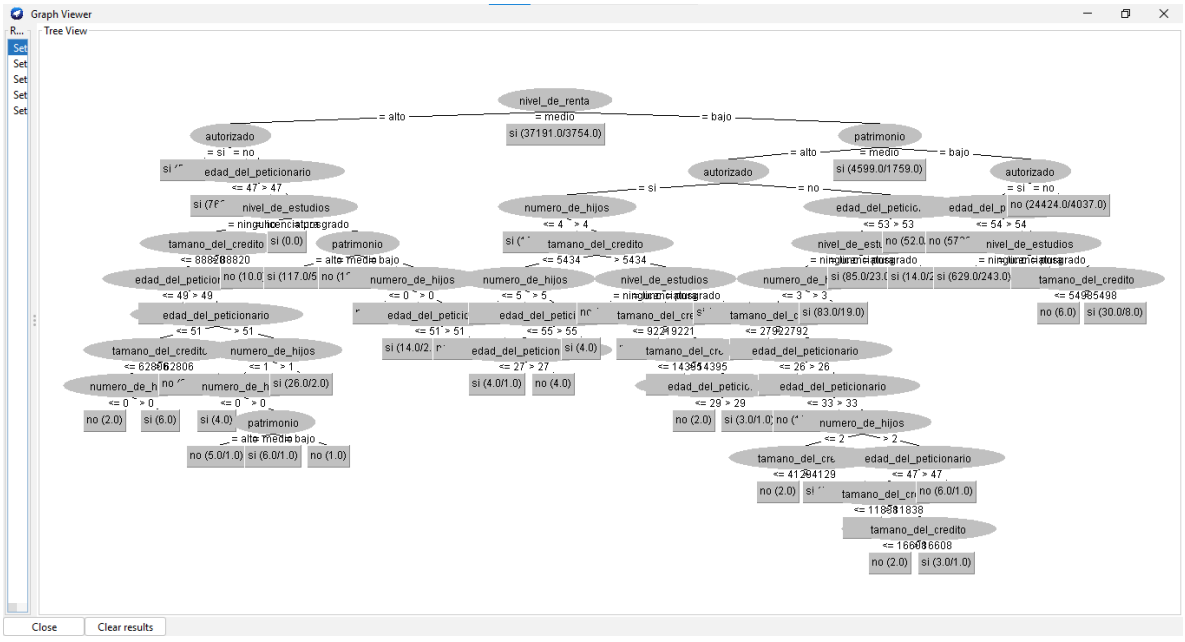

Text Viewer
Result list
06:55:49.845 - Model: J48
06:55:49.942 - Model: J48
06:55:49.969 - Model: J48
06:55:49.977 - Model: J48
06:55:49.983 - Model: J48
Text
| | | | edad_del_peticionario <= 57: si (235.0/76.0)
| | | | edad_del_peticionario > 57
| | | | | numero_de_hijos <= 4
| | | | | | numero_de_hijos <= 3: no (6.0/1.0)
| | | | | | numero_de_hijos > 3: si (4.0)
| | | | | numero_de_hijos > 4: no (8.0)
| patrimonio = medio
| | autorizado = si: si (1366.0/319.0)
| | autorizado = no
| | | edad_del_peticionario <= 25: si (557.0/167.0)
| | | edad_del_peticionario > 25
| | | | nivel_de_estudios = ninguno
| | | | | edad_del_peticionario <= 54: si (1386.0/619.0)
| | | | | edad_del_peticionario > 54: no (445.0/211.0)
| | | | nivel_de_estudios = licenciatura
| | | | | edad_del_peticionario <= 32
| | | | | | numero_de_hijos <= 2
| | | | | | | numero_de_hijos <= 1: no (81.0/36.0)
| | | | | | | numero_de_hijos > 1
| | | | | | | | edad_del_peticionario <= 26: si (9.0/2.0)
| | | | | | | | edad_del_peticionario > 26: no (56.0/22.0)
| | | | | | numero_de_hijos > 2: si (265.0/111.0)
| | | | | edad_del_peticionario > 32: no (289.0/122.0)
| | | | nivel_de_estudios = posgrado: no (128.0/60.0)
| patrimonio = bajo
| | autorizado = si
| | | edad_del_peticionario <= 54: no (5680.0/1835.0)
| | | edad_del_peticionario > 54
| | | | nivel_de_estudios = ninguno: si (607.0/232.0)
| | | | nivel_de_estudios = licenciatura: si (0.0)
| | | | nivel_de_estudios = posgrado
| | | | | tamano_del_credito <= 5410: no (5.0)
| | | | | tamano_del_credito > 5410: si (27.0/6.0)
| | autorizado = no: no (24347.0/3980.0)
Number of Leaves : 45
Size of the tree : 81
Close
Settings
Clear results


Graph Viewer
Tree View
nivel_de_renta
= alto
= medio
= bajo
autorizado
si (37400.0/3777.0)
patrimonio
= si
= no
edad_del_peticionario
= alto
= medio
= bajo
autorizado
autorizado
<= 44
> 44
<= 26
> 26
= si
= no
= si
= no
tamano_del_credito
autorizado
nivel_de_estudios
edad_del_peticionario
no (24304.0/3924.0)
= ninguno
<= 54
> 54
<= 54
> 54
<= 24
> 24
<= 57
> 57
<= 30
> 30
= si
= no
<= 0
> 0
<= 3
> 3
<= 2
> 2
<= 5551
> 5551
<= 50655
> 50655
<= 3374
> 3374
<= 54
> 54
<= 3
> 3
<= 4
> 4
<= 1
> 1
<= 5
> 5
<= 51
> 51
<= 33
> 33
<= 3
> 3
<= 29
> 29
<= 3836
> 3836
<= 6138
> 6138
si (4.0/1.0)
no (11.0/1.0)
no (10.0/1.0)
si (61.0/24.0)
no (8.0/1.0)
si (3.0)
no (7.0/1.0)
si (14.0/4.0)
no (5.0/1.0)
si (47.0/13.0)
si (6.0/1.0)
no (12.0/4.0)


Text Viewer
Result list
06:55:49.845 - Model: J48
06:55:49.942 - Model: J48
06:55:49.969 - Model: J48
06:55:49.977 - Model: J48
06:55:49.983 - Model: J48
Text
| | | nivel_de_estudios = licenciatura
| | | | edad_del_peticionario <= 30: no (24.0/10.0)
| | | | edad_del_peticionario > 30: si (51.0/14.0)
| | | nivel_de_estudios = posgrado: si (25.0/8.0)
| patrimonio = medio
| | autorizado = si: si (1383.0/323.0)
| | autorizado = no
| | | edad_del_peticionario <= 54
| | | | nivel_de_estudios = ninguno: si (1524.0/653.0)
| | | | nivel_de_estudios = licenciatura: si (1102.0/472.0)
| | | | nivel_de_estudios = posgrado
| | | | | numero_de_hijos <= 2
| | | | | | numero_de_hijos <= 1
| | | | | | | edad_del_peticionario <= 29: no (7.0/1.0)
| | | | | | | edad_del_peticionario > 29: si (14.0/4.0)
| | | | | | numero_de_hijos > 1: no (23.0/5.0)
| | | | | numero_de_hijos > 2
| | | | | | numero_de_hijos <= 5
| | | | | | | tamano_del_credito <= 3836: no (5.0/1.0)
| | | | | | | tamano_del_credito > 3836: si (47.0/13.0)
| | | | | | numero_de_hijos > 5
| | | | | | | tamano_del_credito <= 6138: si (6.0/1.0)
| | | | | | | tamano_del_credito > 6138: no (12.0/4.0)
| | | edad_del_peticionario > 54: no (457.0/206.0)
| patrimonio = bajo
| | autorizado = si
| | | edad_del_peticionario <= 54: no (5723.0/1853.0)
| | | edad_del_peticionario > 54
| | | | nivel_de_estudios = ninguno: si (602.0/232.0)
| | | | nivel_de_estudios = licenciatura: si (0.0)
| | | | nivel_de_estudios = posgrado
| | | | | tamano_del_credito <= 5551: no (4.0)
| | | | | tamano_del_credito > 5551: si (28.0/8.0)
| | autorizado = no: no (24304.0/3924.0)
Number of Leaves : 44
Size of the tree : 81
Close
Settings
Clear results


Graph Viewer
Tree View
nivel_de_renta
= alto
= medio
= bajo
autorizado
si (37292.0/3787.0)
patrimonio
= si
= no
= alto
= medio
= bajo
edad_del_peticionario
edad_del_peticionario
autorizado
autorizado
<= 46
> 46
<= 26
> 26
= si
= si
= no
tamano_del_credito
si (87.0/1
autorizado
nivel_de_estudios
no (24396.0/3961.0)
= si
= no
edad_del_peticionario
edad_del_peticionario
edad_del_peti
si (137.
no (5
nivel_de_estudios
<= 55
> 55
<= 56
> 56
<= 32
> 32
nivel_de_estu
si (13.0/1.0)
edad_del_peticionario
numero
si (812.0/
no (272.0/122.0)
si (627.0/230.
tamano_del_credito
= ninguno
= licenciatura
= posgrado
<= 33
> 33
<= 4
> 4
<= 54985498
> 54985498
edad_del_peticion
numero_de_hijos
nivel_de_estudios
numero_de_hijos
numero_de_
no (13.0/1.0)
no (5.0)
si (21.0/7.0)
<= 28
> 28
<= 3
> 3
<= 4
> 4
<= 3
> 3
no (6.0/1.
si (29.0/9.0)
no (2.0)
si
numero_de_hijo
numero_de_h
si (3.0/1.0)
nivel_de_estu
si (50.0/6.
edad_del_peticior
si (4.0)
<= 5
> 5
<= 5
> 5
= ninguno
= licenciatura
= posgrado
<= 57
> 57
no (21.0/7.0)
si (4.0)
si (22.0/10.0)
tamano_del_c.
edad_del_petici
tamano_del_cre
no (7.0)
<= 14484484
<= 37
> 37
<= 81248124
si
numero_de_hij
numero_de_hij
si (2.0)
no (6.0/2.0)
<= 1
> 1
<= 1
> 1
si (6.0/1.
no (19.0/5.0)
no (2.0)
si (6.0/1.0)
Close
Clear results


Text Viewer
Result list
06:55:49.845 - Model: J48
06:55:49.942 - Model: J48
06:55:49.969 - Model: J48
06:55:49.977 - Model: J48
06:55:49.983 - Model: J48
Text
| | | | | | | | nivel_de_estudios = licenciatura
| | | | | | | | | edad_del_peticionario <= 37: si (5.0)
| | | | | | | | | edad_del_peticionario > 37
| | | | | | | | | | numero_de_hijos <= 1: no (2.0)
| | | | | | | | | | numero_de_hijos > 1: si (6.0/1.0)
| | | | | | | | nivel_de_estudios = posgrado: si (12.0/5.0)
| | | | | | | numero_de_hijos > 4: si (50.0/6.0)
| | | | | edad_del_peticionario > 56
| | | | | | numero_de_hijos <= 4
| | | | | | | numero_de_hijos <= 3
| | | | | | | | edad_del_peticionario <= 57
| | | | | | | | | tamano_del_credito <= 8124: si (2.0)
| | | | | | | | | tamano_del_credito > 8124: no (6.0/2.0)
| | | | | | | | edad_del_peticionario > 57: no (7.0)
| | | | | | | numero_de_hijos > 3: si (4.0)
| | | | | numero_de_hijos > 4: no (13.0/1.0)
| patrimonio = medio
| | autorizado = si: si (1369.0/311.0)
| | autorizado = no
| | | nivel_de_estudios = ninguno: si (1988.0/892.0)
| | | nivel_de_estudios = licenciatura
| | | | edad_del_peticionario <= 32: si (812.0/310.0)
| | | | edad_del_peticionario > 32: no (272.0/122.0)
| | | nivel_de_estudios = posgrado: si (137.0/64.0)
| patrimonio = bajo
| | autorizado = si
| | | edad_del_peticionario <= 54: no (5677.0/1856.0)
| | | edad_del_peticionario > 54
| | | | nivel_de_estudios = ninguno: si (627.0/230.0)
| | | | nivel_de_estudios = licenciatura: si (0.0)
| | | | nivel_de_estudios = posgrado
| | | | | tamano_del_credito <= 5498: no (5.0)
| | | | | tamano_del_credito > 5498: si (21.0/7.0)
| | autorizado = no: no (24396.0/3961.0)
Number of Leaves : 41
Size of the tree : 74
Close
Settings
Clear results

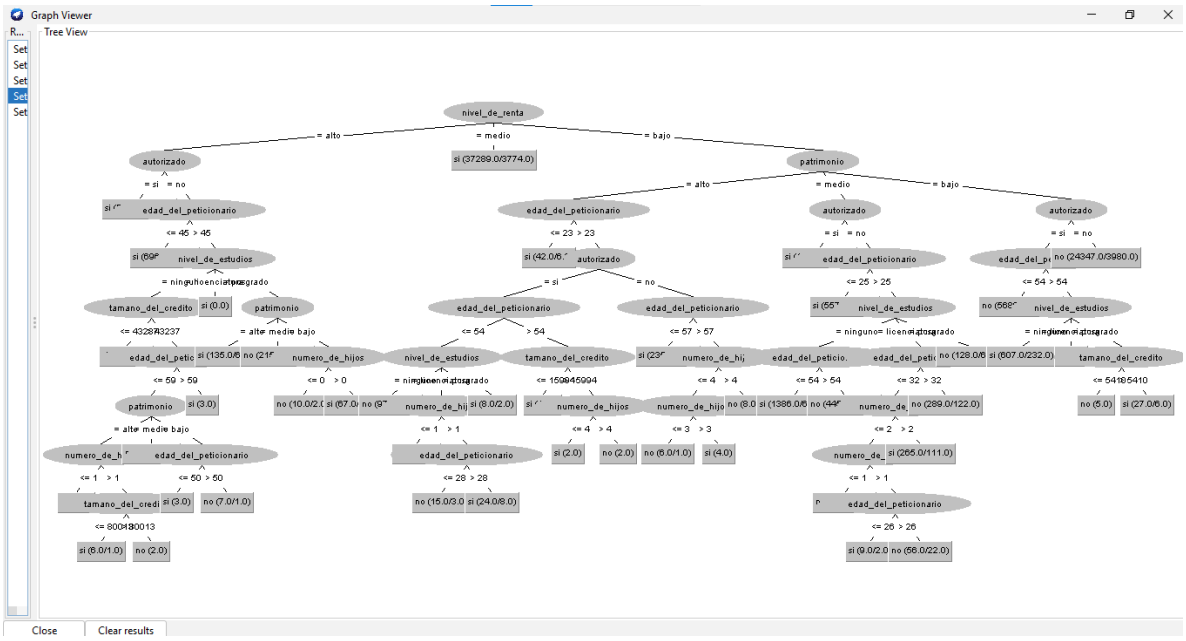

Graph Viewer
Tree View
nivel_de_renta
= alto
= medio
= bajo
si (37191.0/3754.0)
autorizado
patrimonio
= si
= no
= alto
= medio
= bajo
si (4599.0/1759.0)
edad_del_peticionario
<= 47
> 47
numero_de_hijos
<= 4
> 4
edad_del_peticio
<= 53
> 53
<= 54
> 54
no (24424.0/4037.0)
nivel_de_estudios
= ninguno
= licenciatura
= posgrado
si (0.0)
tamano_del_credito
<= 5434
> 5434
<= 49
> 49
<= 51
> 51
<= 55
> 55
<= 3
> 3
si (83.0/19.0)
tamano_del_credito
<= 5498
> 5498
no (6.0)
si (30.0/8.0)
numero_de_hijos
<= 1
> 1
<= 0
> 0
<= 27
> 27
<= 26
> 26
<= 29
> 29
<= 33
> 33
<= 2
> 2
si (26.0/2.0)
no (2.0)
si (6.0)
si (4.0)
si (4.0/1.0)
no (4.0)
si (3.0/1.0)
no (5.0/1.0)
no (1.0)
<= 41284129
<= 11838
<= 16608
Close
Clear results


Text Viewer
Result list
06:55:49.845 - Model: J48
06:55:49.942 - Model: J48
06:55:49.969 - Model: J48
06:55:49.977 - Model: J48
06:55:49.983 - Model: J48
Text
| | | edad_del_peticionario <= 53
| | | | nivel_de_estudios = ninguno
| | | | | numero_de_hijos <= 3
| | | | | | tamano_del_credito <= 2792: si (7.0)
| | | | | | tamano_del_credito > 2792
| | | | | | | edad_del_peticionario <= 26: si (4.0)
| | | | | | | edad_del_peticionario > 26
| | | | | | | | edad_del_peticionario <= 33: no (12.0/1.0)
| | | | | | | | edad_del_peticionario > 33
| | | | | | | | | numero_de_hijos <= 2
| | | | | | | | | | tamano_del_credito <= 4129: no (2.0)
| | | | | | | | | | tamano_del_credito > 4129: si (16.0/3.0)
| | | | | | | | | numero_de_hijos > 2
| | | | | | | | | | edad_del_peticionario <= 47
| | | | | | | | | | | tamano_del_credito <= 11838: si (3.0)
| | | | | | | | | | | tamano_del_credito > 11838
| | | | | | | | | | | | tamano_del_credito <= 16608: no (2.0)
| | | | | | | | | | | | tamano_del_credito > 16608: si (3.0/1.0)
| | | | | | | | | | edad_del_peticionario > 47: no (6.0/1.0)
| | | | | numero_de_hijos > 3: si (83.0/19.0)
| | | | nivel_de_estudios = licenciatura: si (85.0/23.0)
| | | | nivel_de_estudios = posgrado: si (14.0/2.0)
| | | edad_del_peticionario > 53: no (52.0/23.0)
| | patrimonio = medio: si (4599.0/1759.0)
| | patrimonio = bajo
| | | autorizado = si
| | | | edad_del_peticionario <= 54: no (5729.0/1874.0)
| | | | edad_del_peticionario > 54
| | | | | nivel_de_estudios = ninguno: si (629.0/243.0)
| | | | | nivel_de_estudios = licenciatura: si (0.0)
| | | | | nivel_de_estudios = posgrado
| | | | | | tamano_del_credito <= 5498: no (6.0)
| | | | | | tamano_del_credito > 5498: si (30.0/8.0)
| | | autorizado = no: no (24424.0/4037.0)
Number of Leaves : 50
Size of the tree : 91
Close
Settings
Clear results

Post procesamiento

Interpretación de uno de los patrones generados

Como se puede observar, al tener tanto apego a la realidad y detalle, los patrones han salido bastante robustos, con el fin de facilitarlo, analicemos una parte del más pequeño, que pertenece al análisis de funcionario.



Inicialmente se segmenta tomando en cuenta el nivel de renta, por medio del cual el algoritmo concluyó que, si la persona tiene ingresos promedio, entonces es un funcionario, lo cual aparentemente se acopla a muchos de los ejemplos en los datos, más de un tercio para ser específicos. Si el nivel de renta es alto y además se le autorizó el préstamo a la persona, también determina que es un funcionario, en cambio, si el préstamo no fue autorizado, intenta determinarlo por su edad. Esto de cierta manera coincide con las reglas impuestas para la generación de los datos.

De igual forma, en esa rama del árbol descienden muchos nodos, de los cuales la mayoría son razonamientos coherentes, por lo que el patrón se puede catalogar como válido.