Minería de datos usando WEKA

Nombre:
Javier Eduardo Lopez Ontiveros
Materia:
Temas selectos de base de datos
Hora:
11 am a 12pm

Generación del archivo arff

```
        try {
            // Crear el archivo ARFF
            writer = new BufferedWriter(new FileWriter("clientes.arff"));

            // Escribir la sección de encabezado
            writer.write("@relation Registro\n\n");
            writer.write("@attribute Estacion {Primavera, Verano, Otono, Invierno}\n");
            writer.write("@attribute year numeric\n");
            writer.write("@attribute Edad numeric\n");
            writer.write("@attribute NecesitaPermiso {Si, No}\n");
            writer.write("@attribute TienePermiso {Si, No}\n");
            writer.write("@attribute Genero {Masculino, Femenino}\n");
            writer.write("@attribute SeCaso {Si, No}\n");
            writer.write("@data\n");
            // Escribir la sección de datos
            generaDatos();
            // Cerrar el archivo
            writer.close();

            System.out.println("Archivo ARFF generado exitosamente.");
        } catch (IOException e) {
            e.printStackTrace();
        }
```

Se genera la estructura del archivo arff y se llama al metodo generaDatos para generar los ejemplos ejemplo

Generación de la tabla de base de datos

```
CREATE TABLE Personas (
  Estacion VARCHAR(20),
  año INT,
  Edad INT,
  NecesitaPermiso VARCHAR(2),
  TienePermiso VARCHAR(2),
  Genero VARCHAR(10),
  SeCaso VARCHAR(2)
);
```

se crea la tabla en sql para poder llenarla

```
        try {
            con = DriverManager.getConnection(

"jdbc:sqlserver://localhost:1433;encrypt=true;databaseName=Registrocivil;TrustServerCertificate
=true;",
```

```
                user, password);
        } catch (SQLException e) {
            e.printStackTrace();
        }
```
se establece conexión a la base de datos

Metodo generaDatos

```
    private static void generaDatos() {
        String[] estaciones = { "Primavera", "Verano", "Otono", "Invierno" };
        for (int i = 0; i < 2000; i++) {
            StringBuilder ejemplo = new StringBuilder();
            double num = Math.random();
            int edad = (int) ((Math.random() * 51) + 9);
            ejemplo.append(estaciones[(int) (Math.random() * 4)] + ", ");
            ejemplo.append("" + (int) ((Math.random() * 3) + 2020) + ", ");
            ejemplo.append("" + edad + ", ");
            if (num <= 0.95)// se establece el margen de error
                ejemplo.append((edad < 18 ? "Si," + (Math.random() >= 0.5 ? "Si" : "No") : "No, No") +
", ");
            // si la edad es menor a 18 necesita permiso
            else// en caso de que la persona no ocupe permiso se le asigna no tiene permiso
            {// caso de falla o corrupcion
                ejemplo.append((Math.random() >= 0.5 ? "Si" : "No") + ", ");
                ejemplo.append((Math.random() >= 0.5 ? "Si" : "No") + ", ");
            }
            ejemplo.append((Math.random() >= 0.5 ? "Masculino" : "Femenino") + ", ");
            ejemplo.append(Math.random() >= 0.5 ? "Si" : "No");
            insertaTabla(ejemplo);
            ejemplo.append("\n");
            insertaARFF(ejemplo);
        }
    }
```

Se utiliza el mismo método para generar los datos del archivo arff y la tabla sql, este metodo crea 2000 ejemplos y en cada iteración se usan un stringbuilder que es la cadena que se va a agregar, se genera aleatoriamente la edad, la estación y el año del registro y se va agregando al stringbuilder, también se establece un rango de error el cual es de 95% correcto, dentro de este if se valida que la edad sea menor a 18, si es menor significa que SI necesita permiso y se usa otra condición para saber si tiene permiso o no, en caso de que la edad sea mayor a 18 se le asigna no ocupa permiso y no tiene permiso,en caso de que sea margen de error se le puede asignar si ocupa permiso y tiene permiso sin importar la edad, al tener la cadena completa se manda al metodo de inserta tabla y inserta arff

Inserta tabla

```
    private static void insertaTabla(StringBuilder ejemplo) {
```

```
        ejemplo.replace(0, ejemplo.length(),
                ejemplo.toString().replaceAll("([a-zA-Z]+)", "'$1'"));
        try {
            Statement stmt = con.createStatement();
            stmt.executeUpdate("insert into Personas values(" + ejemplo + ")");
        } catch (SQLException e) {
            e.printStackTrace();
        }
    }
```

se agregan comillas a los datos que lo requieren y se crea un statement con la conexión y se inserta en la tabla personas

	Estacion	año	Edad	NecesitaPermiso	TienePermiso	Genero	SeCaso
1	Invierno	2021	42	No	No	Masculino	Si
2	Verano	2020	39	No	No	Femenino	Si
3	Invierno	2021	44	No	No	Femenino	Si
4	Primavera	2022	44	No	No	Femenino	No
5	Primavera	2020	50	No	No	Masculino	Si
6	Invierno	2020	43	No	No	Femenino	No
7	Primavera	2020	34	No	No	Femenino	Si
8	Primavera	2022	36	No	No	Femenino	Si
9	Invierno	2022	12	Si	Si	Femenino	No
10	Verano	2022	54	No	No	Femenino	Si
11	Primavera	2020	27	No	No	Femenino	No
12	Invierno	2020	35	No	No	Femenino	No
13	Primavera	2022	52	No	No	Masculino	No
14	Otono	2021	23	No	No	Femenino	Si
15	Invierno	2021	23	No	No	Masculino	No
16	Invierno	2022	37	No	No	Masculino	No
17	Primavera	2020	56	No	No	Femenino	Si
18	Invierno	2022	22	No	No	Masculino	Si
19	Primavera	2021	15	Si	No	Masculino	Si

Inserta ARFF

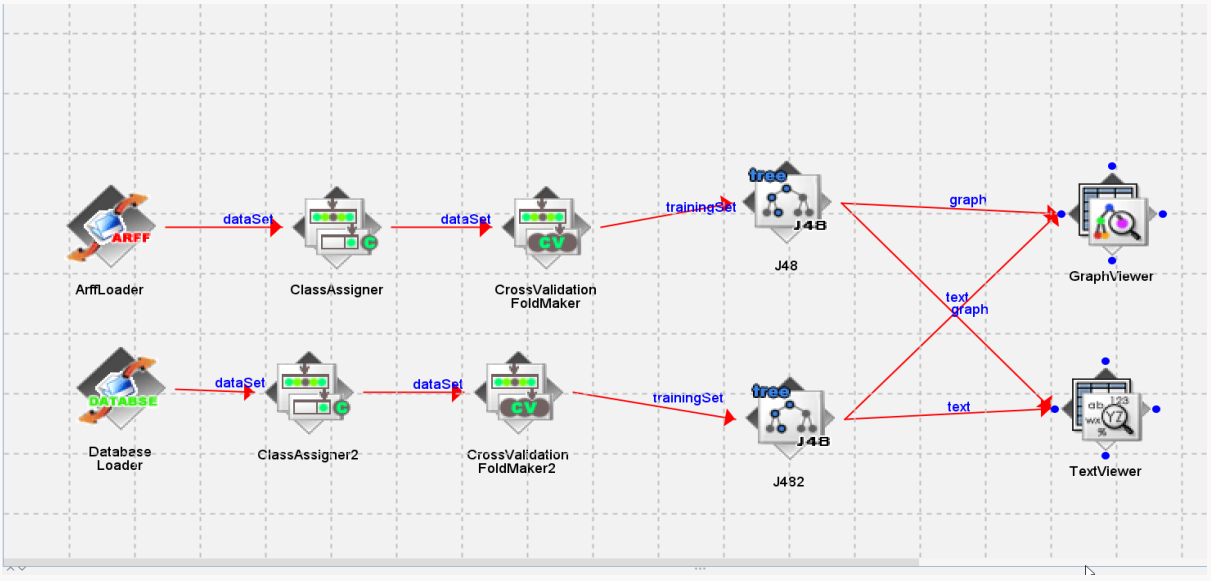
```
    private static void insertaARFF(StringBuilder ejemplo) {
        try {
            writer.write("" + ejemplo);
        } catch (IOException e) {
            e.printStackTrace();
        }
    }
```

se escribe en el documento el nuevo ejemplo

@relation Registro

@attribute Estacion {Primavera, Verano, Otono, Invierno}
@attribute year numeric
@attribute Edad numeric
@attribute NecesitaPermiso {Si, No}
@attribute TienePermiso {Si, No}
@attribute Genero {Masculino, Femenino}
@attribute SeCaso {Si, No}
@data
'Invierno', 2021, 42, 'No', 'No', 'Masculino', 'Si'
'Verano', 2020, 39, 'No', 'No', 'Femenino', 'Si'
'Invierno', 2021, 44, 'No', 'No', 'Femenino', 'Si'
'Primavera', 2022, 44, 'No', 'No', 'Femenino', 'No'
'Primavera', 2020, 50, 'No', 'No', 'Masculino', 'Si'
'Invierno', 2020, 43, 'No', 'No', 'Femenino', 'No'
'Primavera', 2020, 34, 'No', 'No', 'Femenino', 'Si'
'Primavera', 2022, 36, 'No', 'No', 'Femenino', 'Si'
'Invierno', 2022, 12, 'Si','Si', 'Femenino', 'No'
'Verano', 2022, 54, 'No', 'No', 'Femenino', 'Si'
'Primavera', 2020, 27, 'No', 'No', 'Femenino', 'No'
'Invierno', 2020, 35, 'No', 'No', 'Femenino', 'No'
'Primavera', 2022, 52, 'No', 'No', 'Masculino', 'No'
'Otono', 2021, 23, 'No', 'No', 'Femenino', 'Si'
'Invierno', 2021, 23, 'No', 'No', 'Masculino', 'No'
'Invierno', 2022, 37, 'No', 'No', 'Masculino', 'No'
'Primavera', 2020, 56, 'No', 'No', 'Femenino', 'Si'

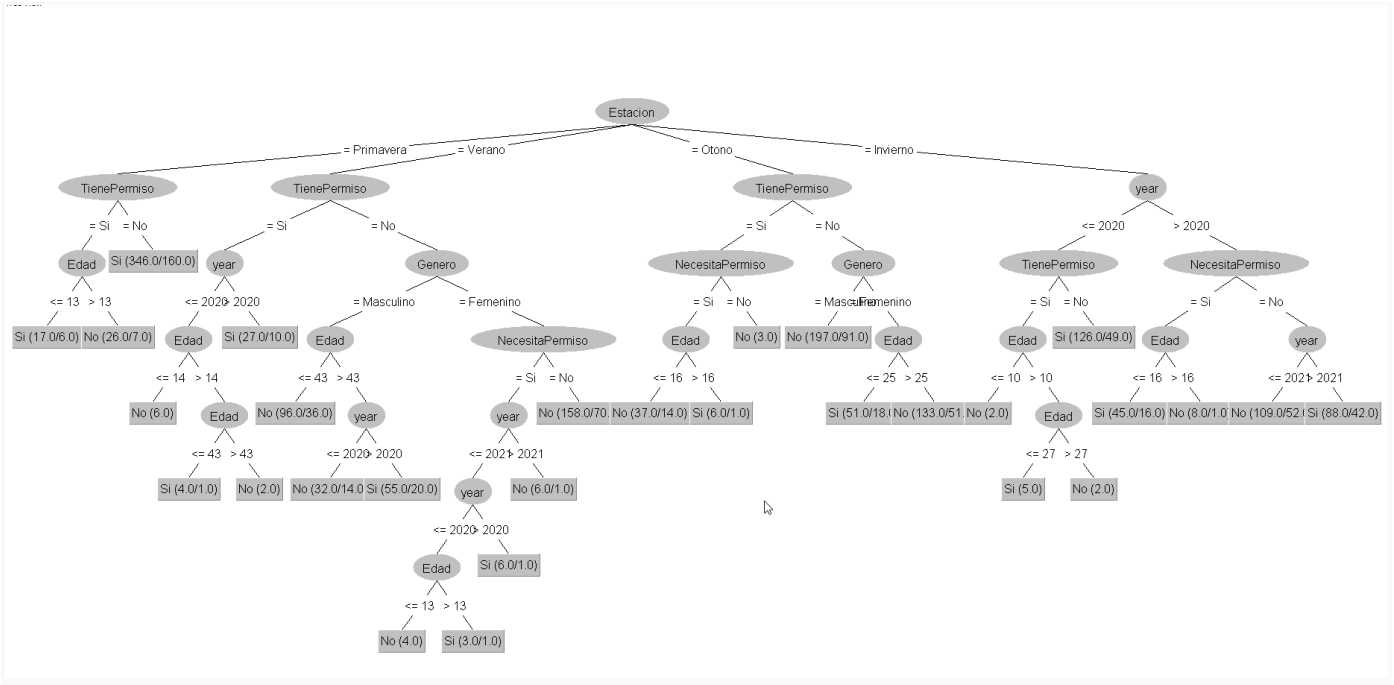
Flujo de conocimiento

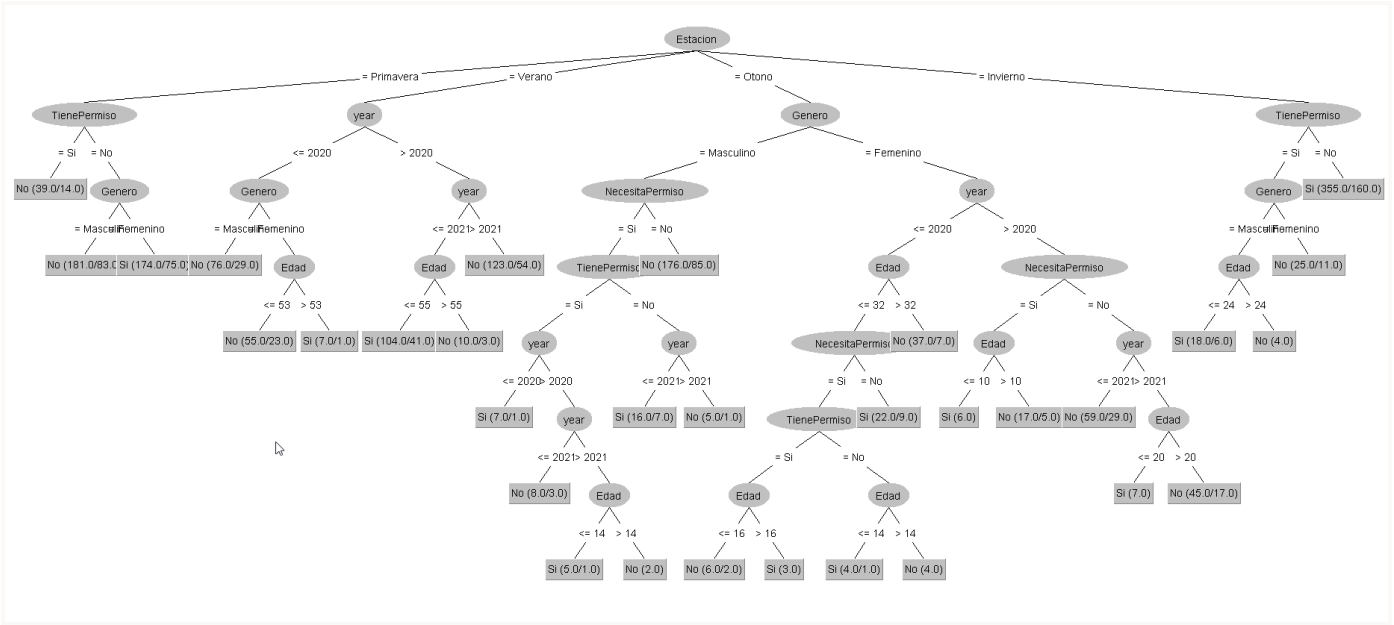


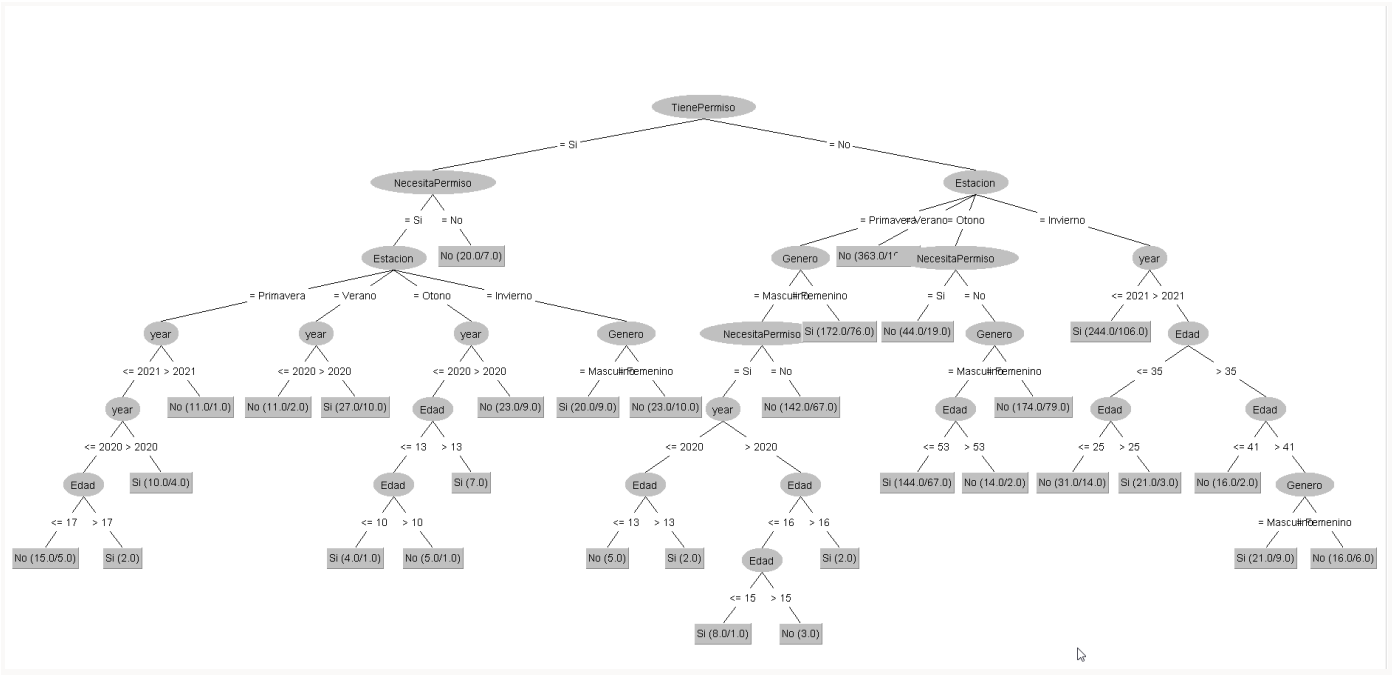
El primer componente es el loader, el cual carga los datos desde la tabla y el archivo arff, después sigue el class asignar este componente lo que hace es determinar en qué atributo va a

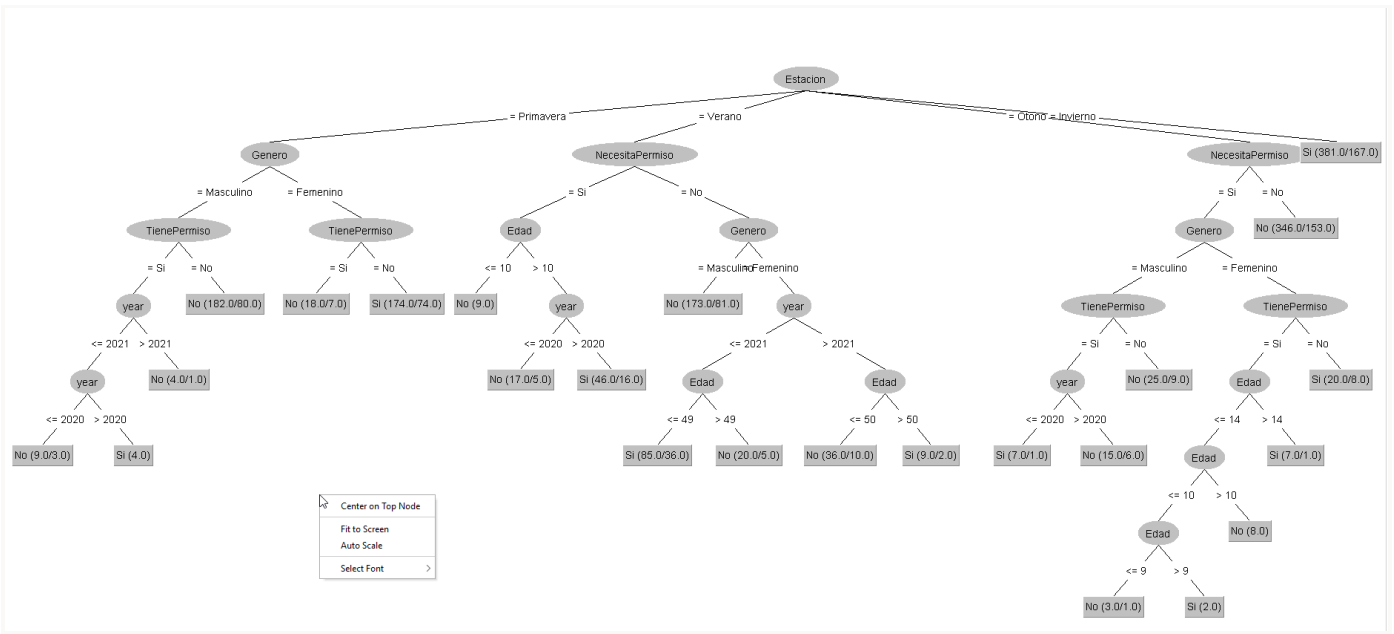
terminar el árbol, El componente Cross Validation es utilizado para determinar cuántos patrones se van generar, este se asigna al algoritmo J48 que es el que se encarga de generar los patrones, finalmente se muestran los patrones con un Graphviewer y un textView

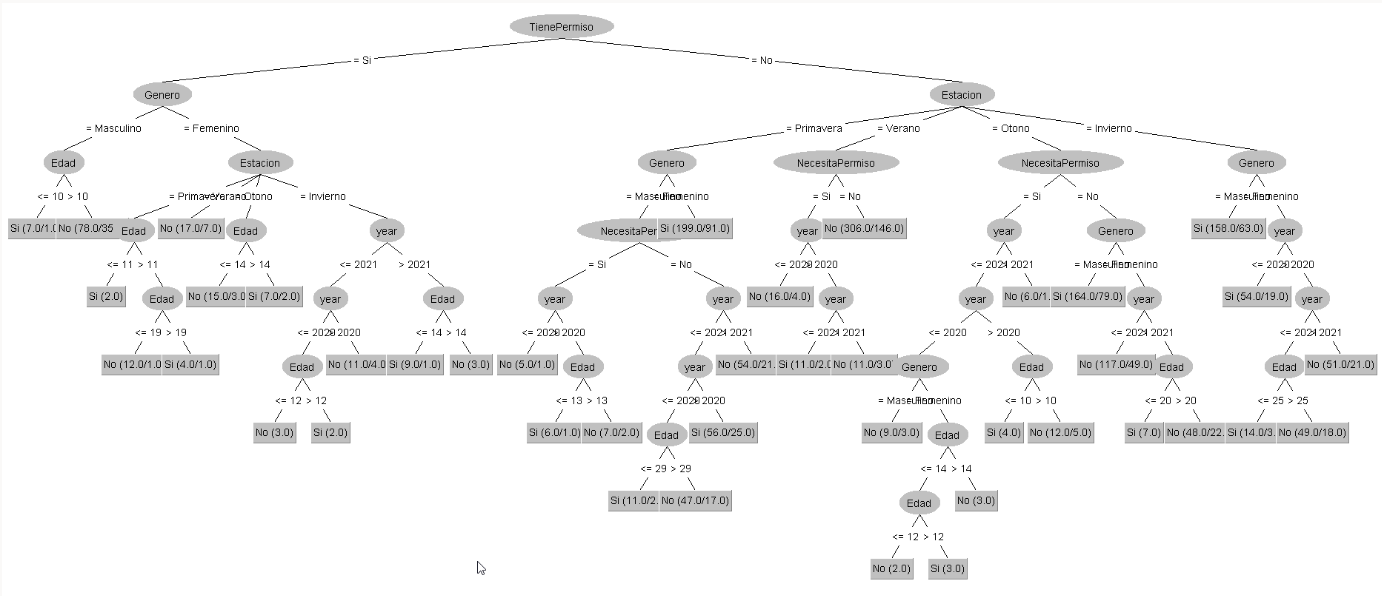
Patrones Generados con el atributo de finalización por “SeCaso”



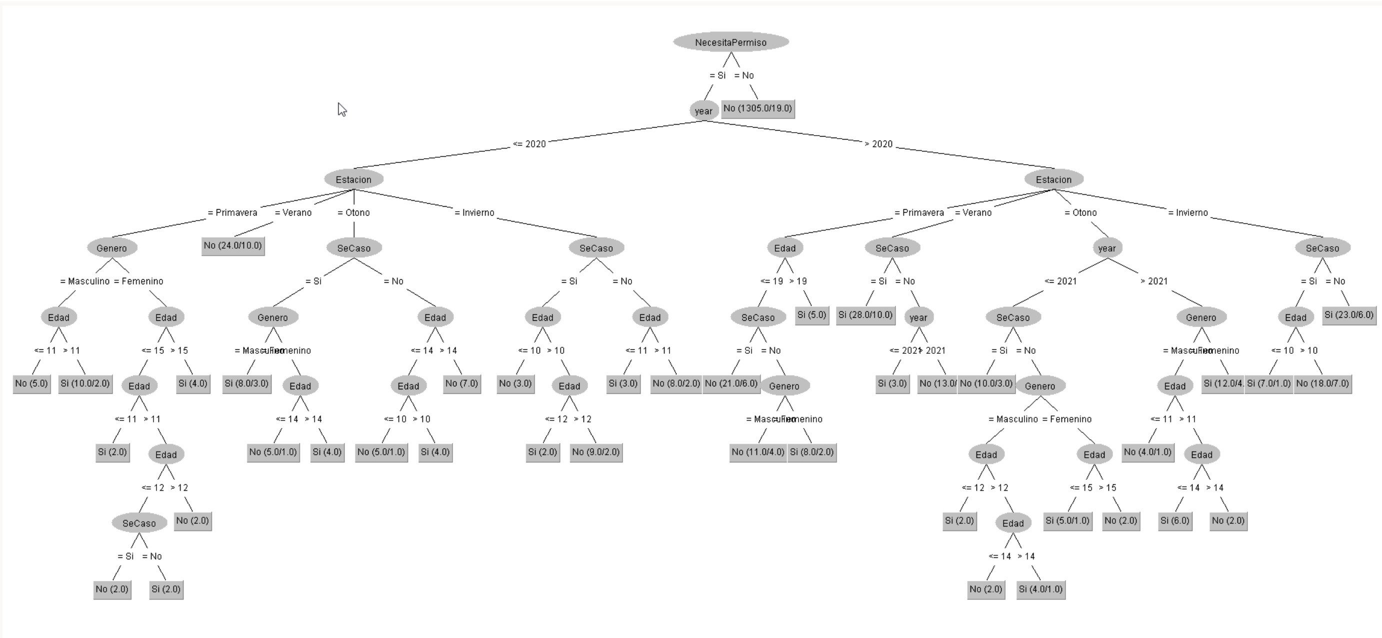


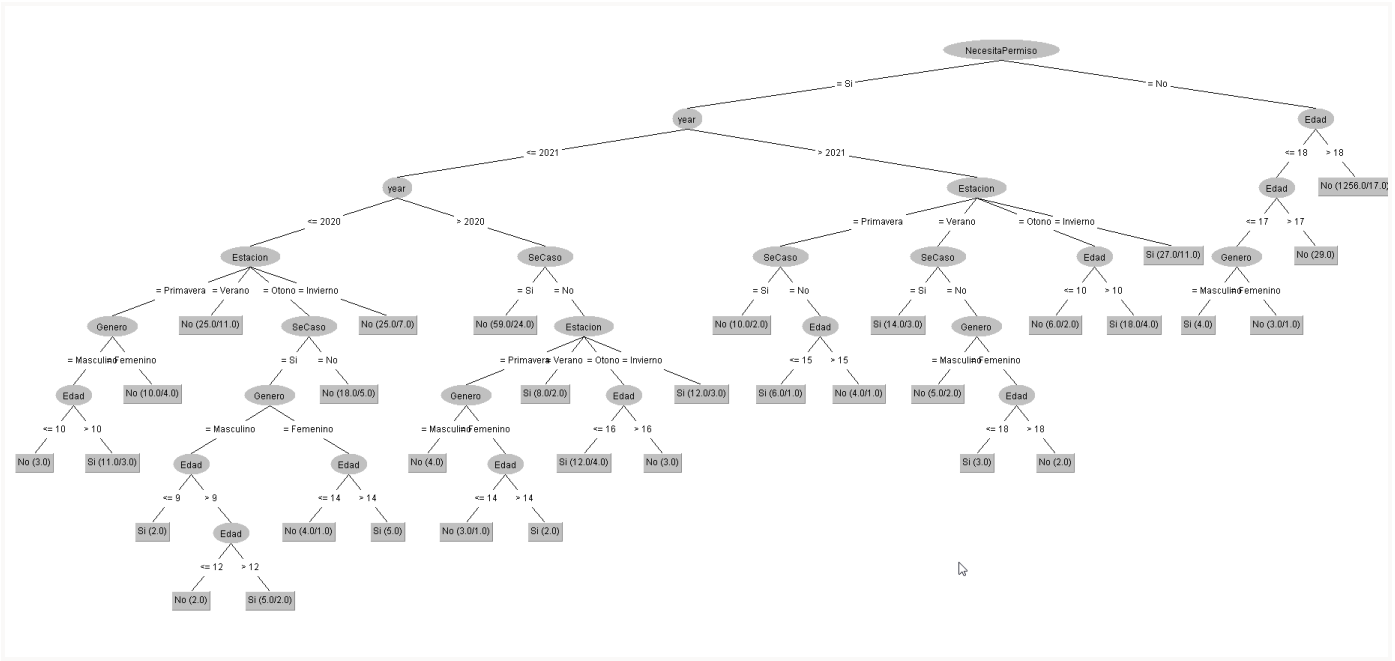

TienePermiso
= Si
= No
NecesitaPermiso
Estacion
= Si
= No
= Primavera
= Verano
= Otono
= Invierno
No (20.0/7.0)
year
Genero
NecesitaPermiso
Genero
year
<= 2021 > 2021
<= 2020 > 2020
<= 2020 > 2020
<= 2020 > 2020
= Masculino
= Femenino
= Si
= No
= Si
= No
= Masculino
= Femenino
<= 2021 > 2021
No (11.0/1.0)
No (11.0/2.0)
Si (27.0/10.0)
Edad
No (23.0/9.0)
Si (20.0/9.0)
No (23.0/10.0)
Si (172.0/76.0)
No (44.0/19.0)
Si (244.0/106.0)
Edad
<= 35
> 35
<= 2020 > 2020
<= 13
> 13
<= 2020
> 2020
<= 53
> 53
<= 25
> 25
<= 41
> 41
Edad
Si (10.0/4.0)
Edad
Si (7.0)
Edad
Edad
Edad
Si (144.0/67.0)
No (14.0/2.0)
No (31.0/14.0)
Si (21.0/3.0)
No (16.0/2.0)
Genero
<= 17
> 17
<= 10
> 10
<= 13
> 13
<= 16
> 16
= Masculino
= Femenino
No (15.0/5.0)
Si (2.0)
Si (4.0/1.0)
No (5.0/1.0)
No (5.0)
Si (2.0)
Edad
Si (2.0)
Si (21.0/9.0)
No (16.0/6.0)
<= 15
> 15
Si (8.0/1.0)
No (3.0)

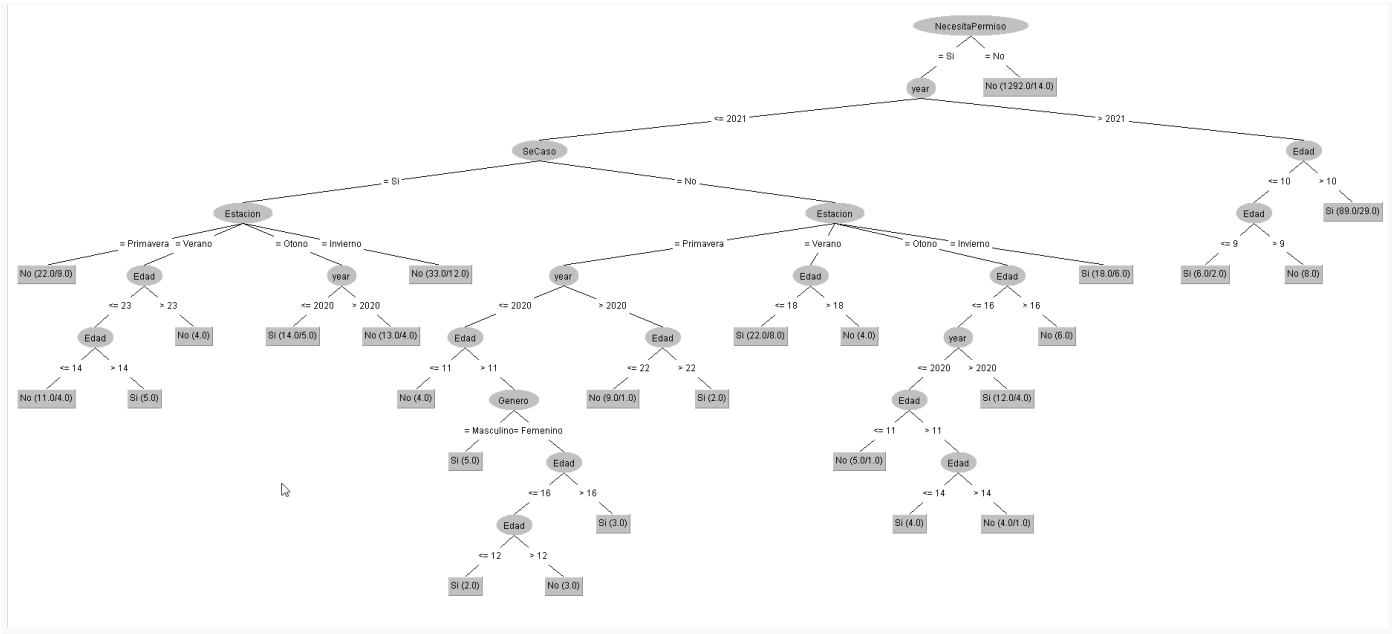

Estacion
= Primavera
= Verano
= Otono
= Invierno
Genero
NecesitaPermiso
NecesitaPermiso
Si (381.0/167.0)
= Masculino
= Femenino
= Si
= No
= Si
= No
TienePermiso
TienePermiso
Edad
Genero
Genero
No (346.0/153.0)
= Si
= No
= Si
= No
= Masculino
= Femenino
year
<= 2021
> 2021
<= 10
> 10
= Masculino
= Femenino
TienePermiso
TienePermiso
No (182.0/80.0)
No (18.0/7.0)
Si (174.0/74.0)
No (9.0)
year
No (173.0/81.0)
year
= Si
= No
= Si
= No
year
No (4.0/1.0)
<= 2020
> 2020
<= 2020
> 2020
<= 2021
> 2021
No (25.0/9.0)
Si (20.0/8.0)
<= 2020
> 2020
No (17.0/5.0)
Si (46.0/16.0)
Edad
Edad
Edad
Edad
<= 14
> 14
No (9.0/3.0)
Si (4.0)
<= 49
> 49
<= 50
> 50
Si (7.0/1.0)
No (15.0/6.0)
Si (7.0/1.0)
Si (85.0/36.0)
No (20.0/5.0)
No (36.0/10.0)
Si (9.0/2.0)
Edad
<= 10
> 10
Edad
No (8.0)
<= 9
> 9
No (3.0/1.0)
Si (2.0)
Center on Top Node
Fit to Screen
Auto Scale
Select Font

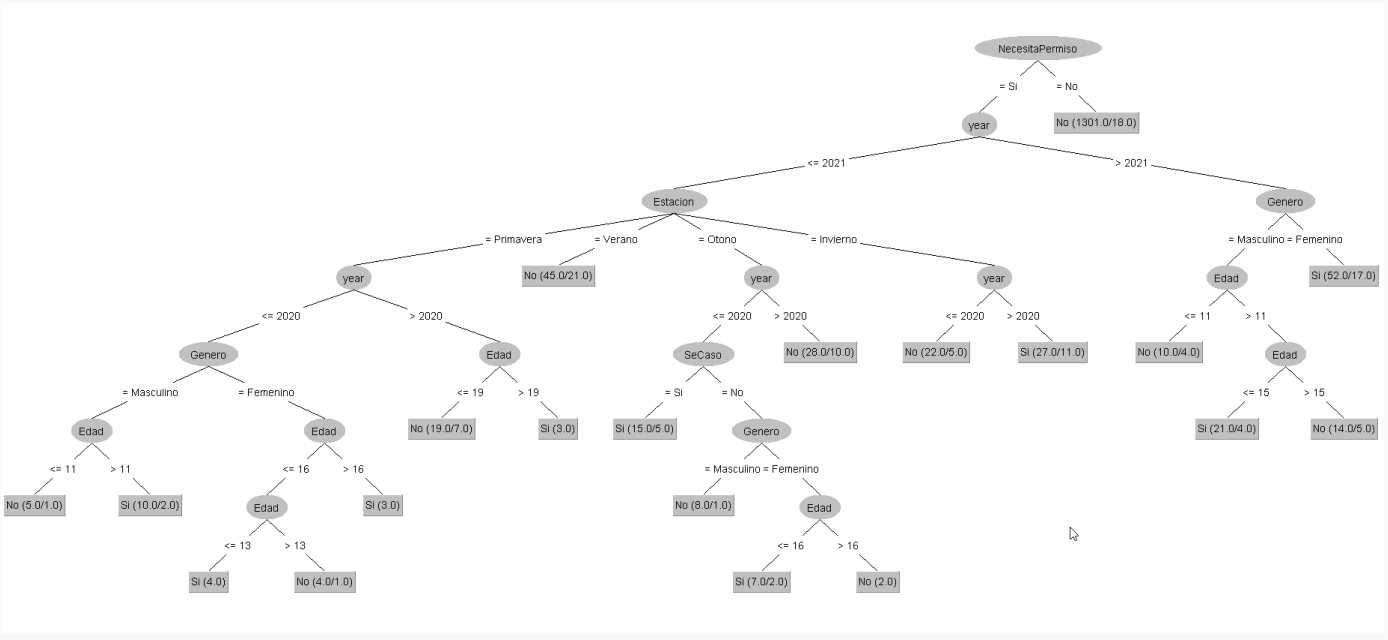


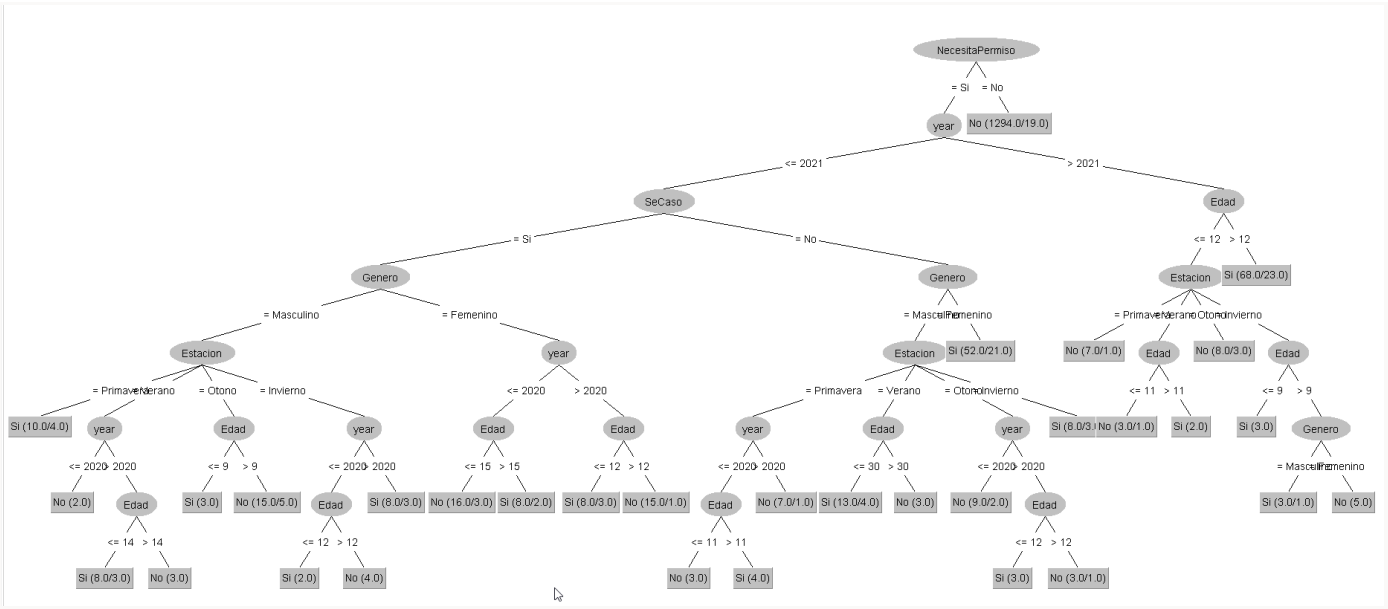
Patrones Generados con el atributo de finalización por “tiene Permiso”



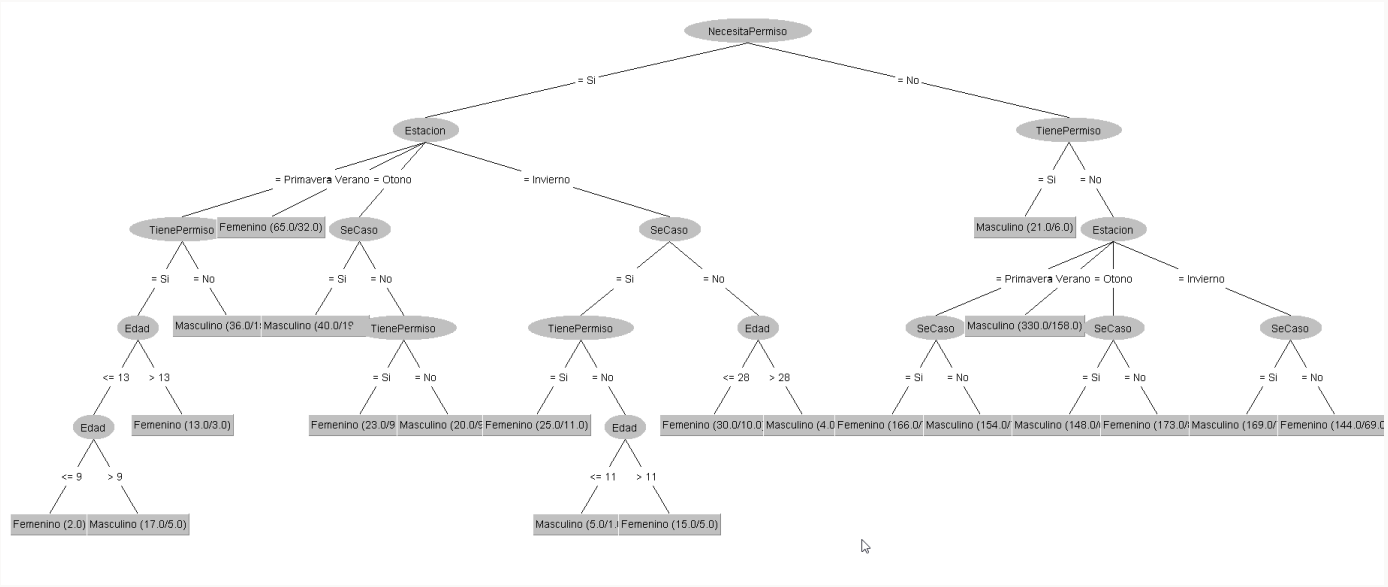

NecesitaPermiso
= Si
= No
year
Edad
<= 2021
> 2021
<= 18
> 18
year
Estacion
Edad
No (1256.0/17.0)
<= 2020
> 2020
= Primavera
= Verano
= Otono
= Invierno
<= 17
> 17
Estacion
SeCaso
SeCaso
SeCaso
Edad
Si (27.0/11.0)
Genero
No (29.0)
= Primavera
= Verano
= Otono = Invierno
= Si
= No
= Si
= No
= Si
= No
<= 10
> 10
= Masculino
= Femenino
Genero
No (25.0/11.0)
SeCaso
No (25.0/7.0)
No (59.0/24.0)
Estacion
No (10.0/2.0)
Edad
Si (14.0/3.0)
Genero
No (6.0/2.0)
Si (18.0/4.0)
Si (4.0)
No (3.0/1.0)
= Masculino
= Femenino
= Si
= No
= Primavera
= Verano
= Otono = Invierno
<= 15
> 15
= Masculino
= Femenino
Edad
No (10.0/4.0)
Genero
No (18.0/5.0)
Genero
Si (8.0/2.0)
Edad
Si (12.0/3.0)
Si (6.0/1.0)
No (4.0/1.0)
No (5.0/2.0)
Edad
<= 10
> 10
= Masculino
= Femenino
= Masculino
= Femenino
<= 16
> 16
<= 18
> 18
No (3.0)
Si (11.0/3.0)
Edad
Edad
No (4.0)
Edad
Si (12.0/4.0)
No (3.0)
Si (3.0)
Si (2.0)
<= 9
> 9
<= 14
> 14
<= 14
> 14
Si (2.0)
Edad
No (4.0/1.0)
Si (5.0)
No (3.0/1.0)
Si (2.0)
<= 12
> 12
No (2.0)
Si (5.0/2.0)

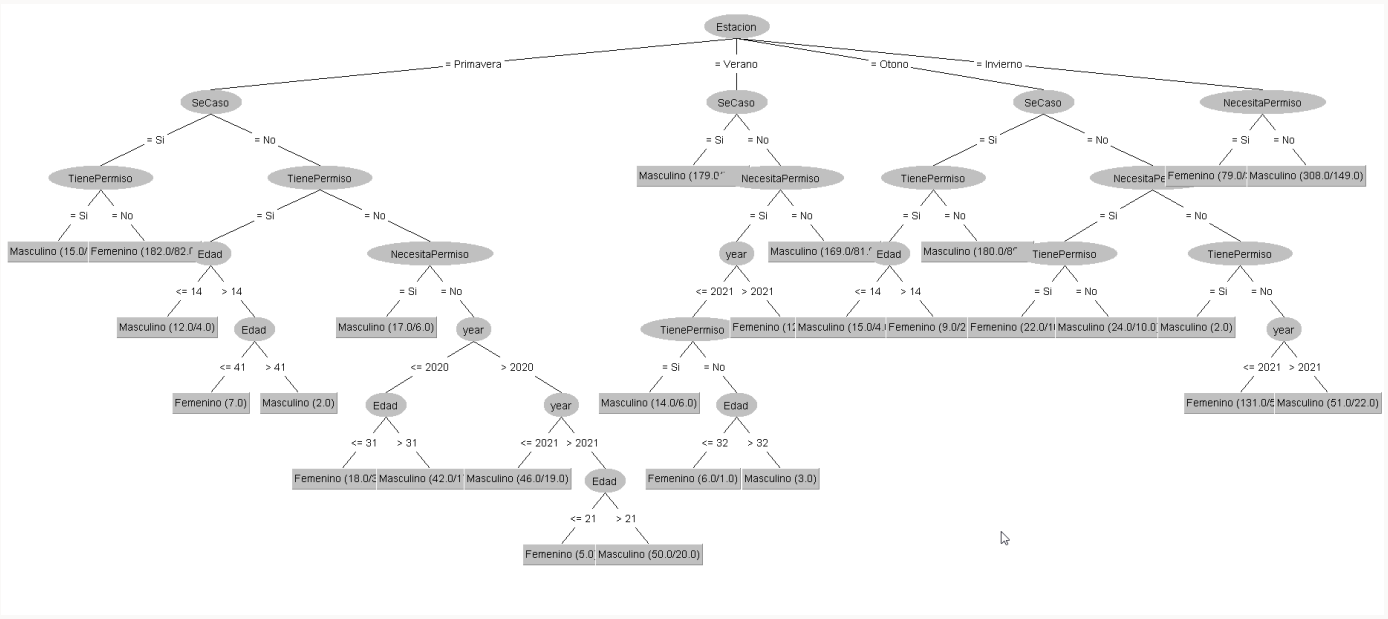

NecesitaPermiso
= Si
= No
year
No (1292.0/14.0)
<= 2021
> 2021
SeCaso
Edad
= Si
= No
<= 10
> 10
Estacion
Estacion
Edad
Si (89.0/29.0)
= Primavera
= Verano
= Otono
= Invierno
= Primavera
= Verano
= Otono
= Invierno
<= 9
> 9
No (22.0/9.0)
Edad
year
No (33.0/12.0)
year
Edad
Edad
Si (18.0/6.0)
Si (6.0/2.0)
No (8.0)
<= 23
> 23
<= 2020
> 2020
<= 2020
> 2020
<= 18
> 18
<= 16
> 16
Edad
No (4.0)
Si (14.0/5.0)
No (13.0/4.0)
Edad
Edad
Si (22.0/8.0)
No (4.0)
Edad
No (6.0)
<= 14
> 14
<= 11
> 11
<= 22
> 22
<= 2020
> 2020
No (11.0/4.0)
Si (5.0)
No (4.0)
Genero
No (9.0/1.0)
Si (2.0)
Edad
Si (12.0/4.0)
= Masculino= Femenino
<= 11
> 11
Si (5.0)
Edad
No (5.0/1.0)
Edad
<= 16
> 16
<= 14
> 14
Si (3.0)
Edad
Si (4.0)
No (4.0/1.0)
<= 12
> 12
Si (2.0)
No (3.0)

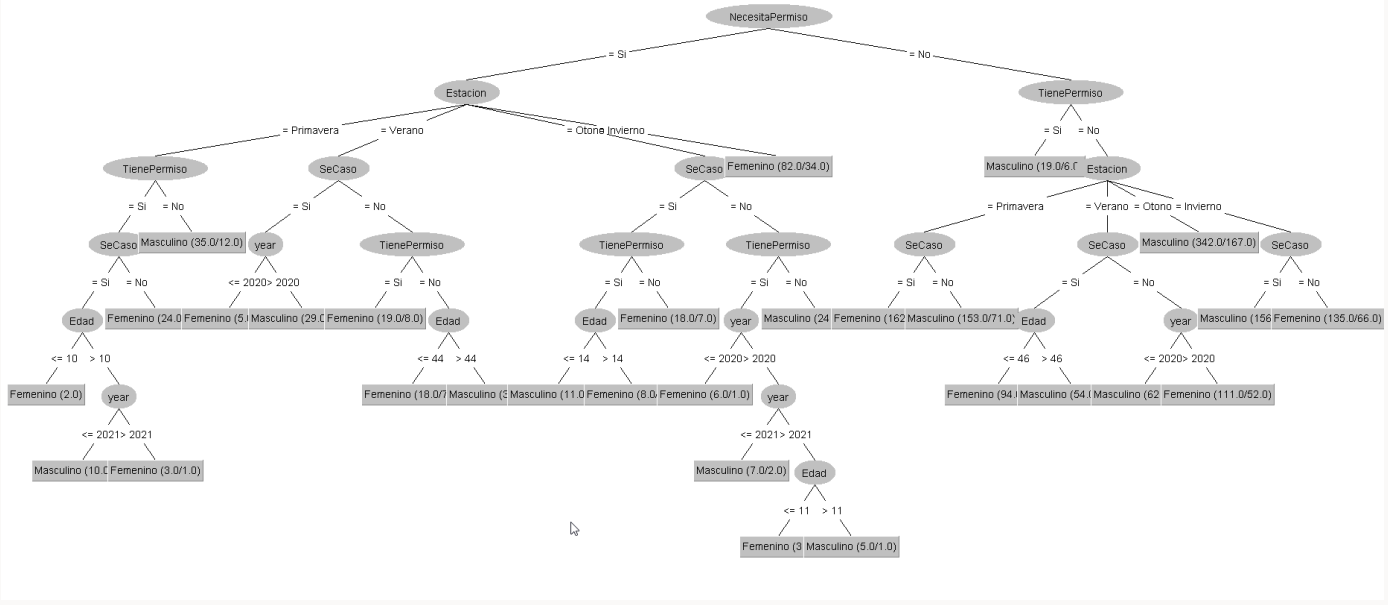

NecesitaPermiso
= Si
= No
No (1301.0/18.0)
year
<= 2021
> 2021
Estacion
Genero
= Primavera
= Verano
= Otono
= Invierno
= Masculino
= Femenino
year
No (45.0/21.0)
year
year
Edad
Si (52.0/17.0)
<= 2020
> 2020
<= 2020
> 2020
<= 2020
> 2020
<= 11
> 11
Genero
Edad
SeCaso
No (28.0/10.0)
No (22.0/5.0)
Si (27.0/11.0)
No (10.0/4.0)
Edad
= Masculino
= Femenino
<= 19
> 19
= Si
= No
<= 15
> 15
Edad
Edad
No (19.0/7.0)
Si (3.0)
Si (15.0/5.0)
Genero
Si (21.0/4.0)
No (14.0/5.0)
<= 11
> 11
<= 16
> 16
= Masculino
= Femenino
No (5.0/1.0)
Si (10.0/2.0)
Edad
Si (3.0)
No (8.0/1.0)
Edad
<= 13
> 13
<= 16
> 16
Si (4.0)
No (4.0/1.0)
Si (7.0/2.0)
No (2.0)

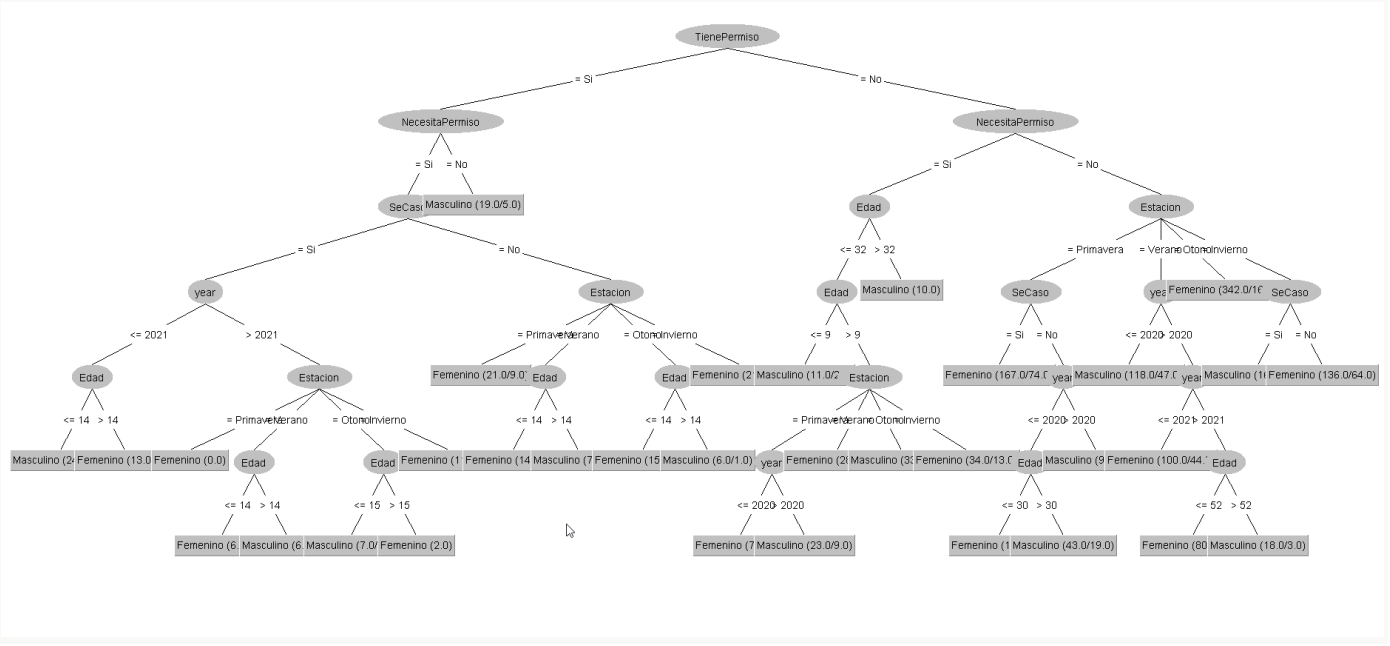

NecesitaPermiso
= Si
= No
year
No (1294.0/19.0)
<= 2021
> 2021
SeCaso
Edad
= Si
= No
<= 12
> 12
Genero
Genero
Estacion
Si (68.0/23.0)
= Masculino
= Femenino
= Masculino
= Femenino
= Primavera
= Verano
= Otono
= Invierno
Estacion
year
Estacion
Si (52.0/21.0)
No (7.0/1.0)
Edad
No (8.0/3.0)
Edad
= Primavera
= Verano
= Otono
= Invierno
<= 2020
> 2020
= Primavera
= Verano
= Otono
= Invierno
<= 11
> 11
<= 9
> 9
Si (10.0/4.0)
year
Edad
year
Edad
Edad
year
Edad
year
Si (8.0/3.0)
No (3.0/1.0)
Si (2.0)
Si (3.0)
Genero
<= 2020
> 2020
<= 9
> 9
<= 2020
> 2020
<= 15
> 15
<= 12
> 12
<= 2020
> 2020
<= 30
> 30
<= 2020
> 2020
= Masculino
= Femenino
No (2.0)
Edad
Si (3.0)
No (15.0/5.0)
Edad
Si (8.0/3.0)
No (16.0/3.0)
Si (8.0/2.0)
Si (8.0/3.0)
No (15.0/1.0)
Edad
No (7.0/1.0)
Si (13.0/4.0)
No (3.0)
No (9.0/2.0)
Edad
Si (3.0/1.0)
No (5.0)
<= 14
> 14
<= 12
> 12
<= 11
> 11
<= 12
> 12
Si (8.0/3.0)
No (3.0)
Si (2.0)
No (4.0)
No (3.0)
Si (4.0)
Si (3.0)
No (3.0/1.0)

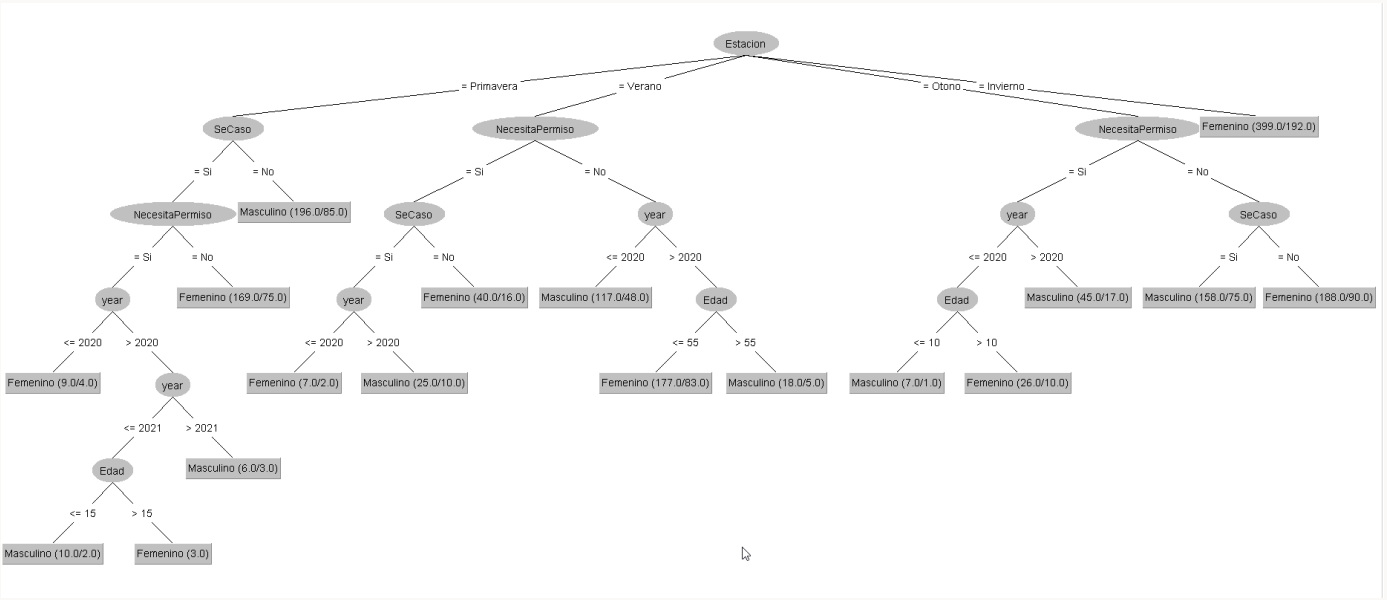
Patrones Generados con el atributo de finalización por “género”



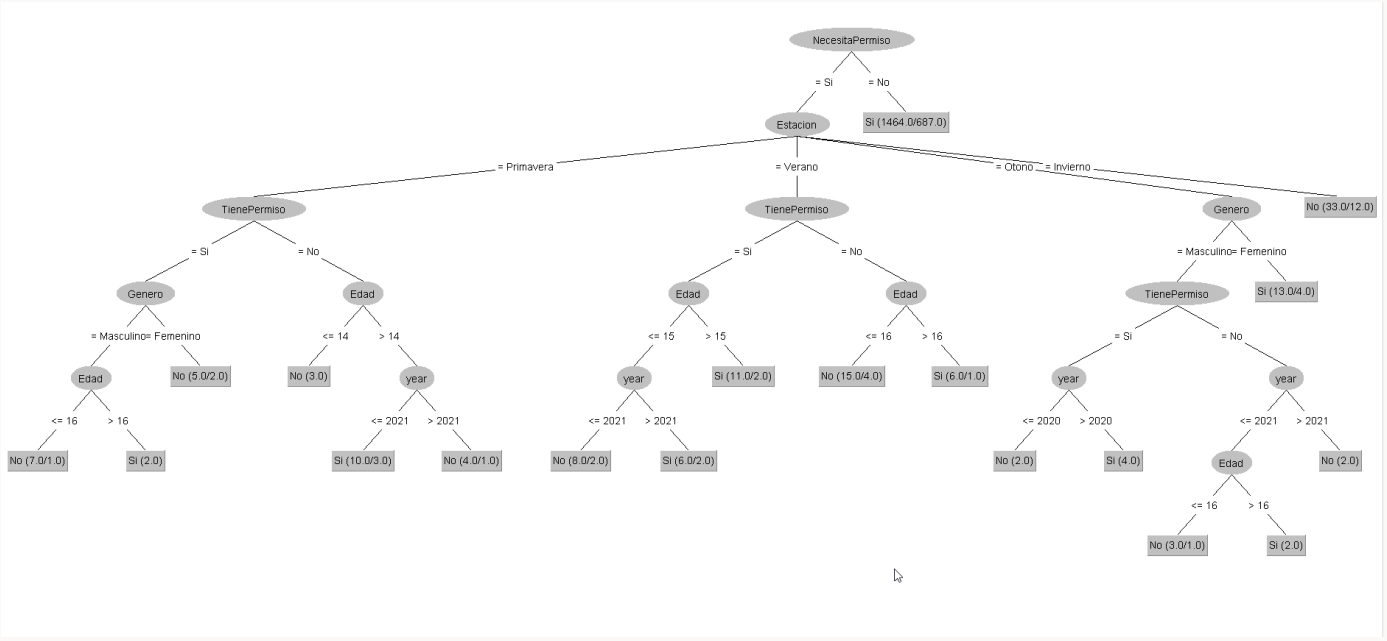



NecesitaPermiso
= Si
= No
Estacion
TienePermiso
= Primavera
= Verano
= Otono
Invierno
= Si
= No
TienePermiso
SeCaso
SeCaso
Femenino (82.0/34.0)
Masculino (19.0/6.
Estacion
= Si
= No
SeCaso
Masculino (35.0/12.0)
year
TienePermiso
TienePermiso
TienePermiso
SeCaso
SeCaso
Masculino (342.0/167.0)
SeCaso
= Si
= No
<= 2020
> 2020
Edad
Femenino (24.0
Femenino (5.
Masculino (29.
Femenino (19.0/8.0)
Edad
Edad
Femenino (18.0/7.0)
year
Masculino (24
Femenino (162
Masculino (153.0/71.0)
Edad
year
Masculino (156
Femenino (135.0/66.0)
<= 10
> 10
<= 44
> 44
<= 14
> 14
<= 2020
> 2020
<= 46
> 46
<= 2020
> 2020
Femenino (2.0)
year
Femenino (18.0/7
Masculino (5
Masculino (11.0
Femenino (8.0
Femenino (6.0/1.0)
year
Femenino (94.
Masculino (54.
Masculino (62
Femenino (111.0/52.0)
<= 2021
> 2021
<= 2021
> 2021
Masculino (10.
Femenino (3.0/1.0)
Masculino (7.0/2.0)
Edad
<= 11
> 11
Femenino (3
Masculino (5.0/1.0)

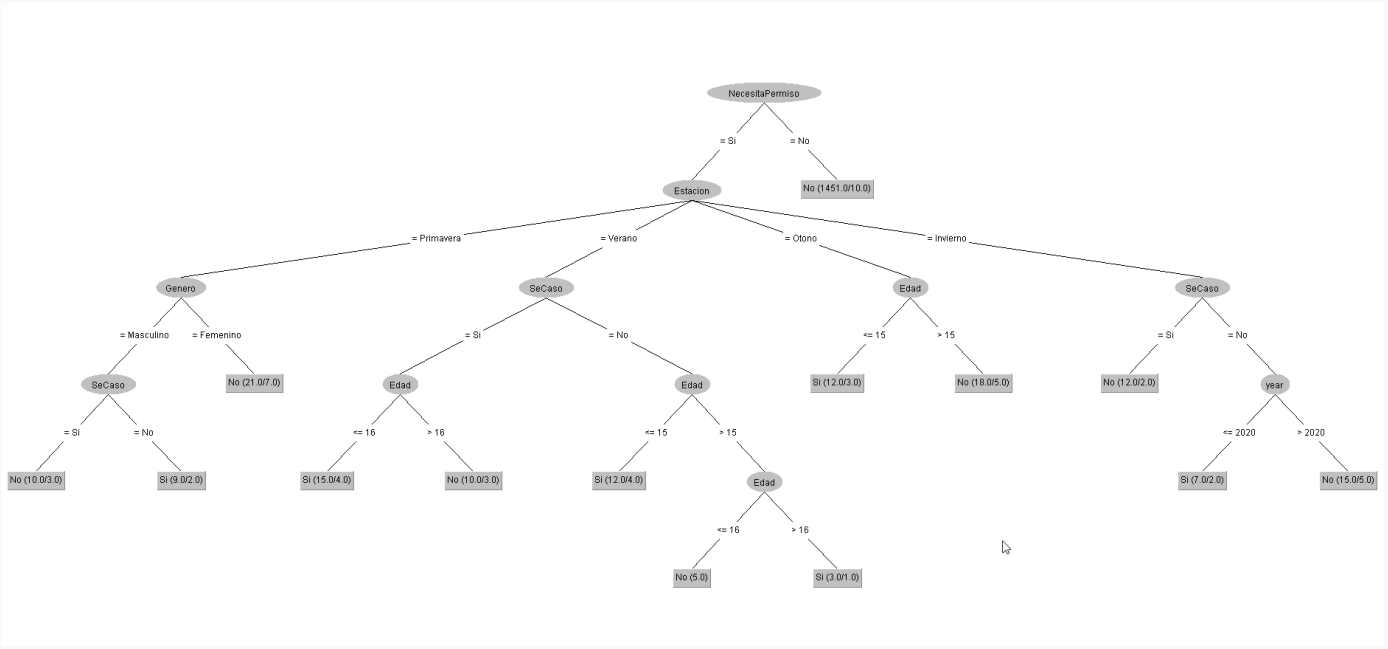

TienePermiso
= Si
= No
NecesitaPermiso
NecesitaPermiso
= Si
= No
SeCas
Masculino (19.0/5.0)
Edad
Estacion
= Si
= No
<= 32
> 32
= Primavera
= Verano
Otono
Invierno
year
Estacion
Edad
Masculino (10.0)
SeCaso
yea
Femenino (342.0/16
SeCaso
<= 2021
> 2021
= Primavera
Verano
= Otono
Invierno
<= 9
> 9
= Si
= No
<= 2020
> 2020
= Si
= No
Edad
Estacion
Femenino (21.0/9.0
Edad
Edad
Femenino (2
Masculino (11.0/2
Estacion
Femenino (167.0/74.
yea
Masculino (118.0/47.
yea
Masculino (1
Femenino (136.0/64.0)
<= 14
> 14
= Primavera
Verano
= Otono
Invierno
<= 14
> 14
<= 14
> 14
= Primavera
Verano
Otono
Invierno
<= 2020
> 2020
<= 2021
> 2021
Masculino (2
Femenino (13.0
Femenino (0.0)
Edad
Edad
Femenino (1
Femenino (14
Masculino (7
Femenino (15
Masculino (6.0/1.0)
year
Femenino (2
Masculino (3
Femenino (34.0/13.0
Edad
Masculino (9
Femenino (100.0/44.
Edad
<= 14
> 14
<= 15
> 15
<= 2020
> 2020
<= 30
> 30
<= 52
> 52
Femenino (6.
Masculino (6.
Masculino (7.0/
Femenino (2.0)
Femenino (7
Masculino (23.0/9.0)
Femenino (1
Masculino (43.0/19.0)
Femenino (80
Masculino (18.0/3.0)



Interpretando un patrón



Este patrón indica que si una persona no necesita permiso, puede que se case en la mayoría de las veces pero si necesita permiso, se pregunta por la estación del año, en este caso en la rama de invierno no se pudieron casar, por lo que no tiene mucho sentido, otra rama es que sea verano y que la persona tenga permiso y que tenga menos de 15 años y el año sea antes de 2022 entonces la persona no se podrá casar



Este patrón determina que si alguien no necesita permiso, entonces no tiene permiso pero si tiene permiso se pregunta por la estación y si la persona se caso y si su edad es menor a 16 entonces si tiene permiso