

Informe Progreso de Detección de Anomalías en Asistencia (Opción 4)

Alumnos:

- Tomás Altamirano
- Gonzalo Petosa

Elegimos trabajar con la opción número 4 debido a que nos pareció más adaptable a nuestra forma de trabajar con **Isolation Forest**.

Primero utilizamos una semilla de aleatoriedad para obtener un mismo resultado cada vez que testeábamos, luego comenzamos a simular los datos de los 100 empleados más los 30 días que estos deberían asistir, en donde el número **1** representa que asistió y el número **0** que faltó, para poder verificar que las anomalías se detectan, inyectamos a drede 3 empleados con anomalías, es decir con al menos 3 inasistencias aseguradas, lo que nos ayuda a verificar los resultados. Lo siguiente que hicimos fué crear un **DataFrame** el cual es una tabla con los días con el valor de asistencia por empleado. Seguido de esto, usamos una expresión lambda para transformar los **0** en asistencias y los **1** en inasistencias ya que facilita el cálculo de la racha de ausencia, luego aplicamos el **Insolation Forest** para comenzar a detectar las anomalías, dejando una contaminación en 0.1 ya que el porcentaje de ausentismo de un empleado ronda el 10%, y luego las predijimos en donde **1** es para normal y **-1** es para anomalía y por consiguiente creamos un DataFrame con estas predicciones, mostramos los resultados de estas para luego graficar sus resultados, que primero vamos a graficar la asistencia de los primeros 5 empleados para ver cómo se distribuyen, seleccionamos los empleados para la visualización y finalmente graficamos las asistencias de estos empleados, en dónde se nos muestra las anomalías que se registraron.

Antes de ejecutar el código se debe instalar mediante consola las siguiente librerías de la siguiente manera:

```
"pip install numpy pandas matplotlib"
```

Nuestro código se encuentra en:

<https://github.com/7oXaL/Plataforma-de-Gestion-de-Recursos-Humanos-con-ERP-y-CRM-con-Inteligencia-Artificial/blob/main/DataCet.py>