

## PRACTICA 02

### ENTREGABLE 1: ANÁLISIS DE TECNOLOGÍAS Y ELECCIÓN DE LA TECNOLOGÍA

El desarrollo de un sistema de clasificación para detectar incongruencias en los resultados de análisis de prevención de riesgos laborales de empresas implica varios pasos que podemos ver a continuación:

#### **1. Análisis de Tecnologías:**

- **Lenguaje de Programación:** Python es una elección sólida debido a su amplia comunidad, bibliotecas de aprendizaje automático y herramientas de análisis de datos.
- **Librerías de Machine Learning:** Para la clasificación, podemos utilizar Scikit-Learn, una biblioteca de aprendizaje automático en Python. Es open source y ampliamente utilizada.
- **Procesamiento y visualización de Datos:** Pandas y NumPy para la manipulación y procesamiento de datos y Matplotlib para visualizar datos.
- **Entorno de Desarrollo:** Jupyter Notebooks o IDEs como VSCode.
- **Herramientas para Modelado:** Scikit-Learn dispone, entre otros, del algoritmo *cosine\_similarity()* ideal para nuestro proyecto.

#### **2. Costes Asociados:**

La mayoría de las tecnologías mencionadas son open source y no tienen costos asociados, excepto que decidamos utilizar servicios en la nube para el almacenamiento de datos o entrenamiento de modelos. En ese caso, no lo hemos necesitado, aunque dependiendo del proyecto, quizás sea necesario utilizar plataformas como AWS que dispone tanto de un nivel gratuito con algunos límites, como niveles no gratuitos.

#### **3. Elección de la Tecnología:**

Dado que se prefiere el uso de tecnologías open source sin costos asociados, Python con Scikit-Learn y las bibliotecas mencionadas es una elección adecuada para este proyecto.

### ENTREGABLE 2: DESARROLLO Y ENTREGA

Esta segunda parte, se ha realizado en notebook dividido en 3 partes:

1. **Preparacion de datos:** eliminar columnas, duplicados...
2. **Análisis de datos y extracción de estadísticas:** riesgos mas frecuentes, menos frecuentes, riesgos por sector, por cliente...
3. **Obtencion de la similitud** entre una evaluacion determinada con todas las demas del mismo sector, usando el algoritmo *cosine\_similarity()*

Para ver el código y los resultados con detalle, acceda al archivo *practica02.ipynb*