Manual de Instalación del Tablero
Universidad de los Andes
Despliegue de Soluciones Analíticas
Grupo 2 - Laura Becerra, Diego Monroy, Robert López y Cesar Porras

Para la correcta instalación del tablero es necesario presentar una descripción del código del tablero en Dash (https://github.com/rmlopezgit/prec/blob/main/Code/Dash_2.py), y le código principal de la API (https://github.com/rmlopezgit/prec/blob/main/Code/app/api.py).

API app/api.py

1. Librerías:

- Se importan diversas librerías como **pandas** para manipulación de datos, herramientas de clustering de **scipy**, herramientas de preprocesamiento y análisis de componentes principales (PCA) de **sklearn**, y otras para la construcción de la API como **fastapi**.

2. Definición de la Ruta y Verificación de Salud:

- Se establece una ruta **/health** que responde con información sobre la salud de la API. Esta ruta se utiliza para verificar que la API esté en funcionamiento correctamente.

3. Ruta de Predicción:

- Se define una ruta **/predict** que acepta datos de entrada según un esquema predefinido.
- Dentro de esta ruta, se realiza un proceso de predicción basado en un modelo de clusterización. Los resultados se almacenan en un DataFrame y se exportan a un archivo Excel.
- La ruta devuelve un diccionario con los resultados de la predicción y metadatos.

4. Proceso de Predicción:

- Se lee la entrada de datos y se combina con información adicional de un archivo Excel.
- Se realiza un análisis de componentes principales (PCA) y se aplica la jerarquía de clústeres.
- Los resultados se almacenan en un DataFrame y se exportan a un archivo Excel llamado "resultados_clusterizacion.xlsx".

5. Respuesta y Resultados:

- La ruta de predicción devuelve un diccionario que incluye las predicciones, posibles errores y la versión del modelo.

6. Comentarios Adicionales:

- Algunos fragmentos de código comentados y no utilizados se han dejado en el código original, pero no se han incluido en la descripción.

Tablero Dash_2.py

1. Importación de Librerías:

- Se importan diversas librerías, incluyendo Dash para la creación de la aplicación web, pandas para la manipulación de datos, plotly para la generación de gráficos interactivos, y otras utilidades**.**

2. Carga de Datos y Creación de la Aplicación Dash:

- Se define una función load_data_2 para cargar los datos desde un archivo Excel y configurar el DataFrame.
- Se inicializa la aplicación Dash.

3. Diseño de la Interfaz de Usuario:

- Se crea la interfaz de usuario utilizando componentes HTML y Dash.
- Se incluyen etiquetas, entradas de texto, botones y gráficos interactivos.

4. Callbacks:

- Se definen callbacks para manejar la interactividad de la aplicación. Por ejemplo, hay un callback para actualizar los gráficos cuando se seleccionan ciertas opciones.

5. Funciones de Actualización de Gráficos y Llamada a la API:

- Se definen funciones para actualizar los gráficos con base en las selecciones del usuario.
- También se define una función llamar_api que realiza una llamada a una API externa cuando se hace clic en un botón.

6. Ejecución de la Aplicación:

- Se ejecuta la aplicación Dash en el servidor local en el puerto 8050.

7. Funcionalidades Principales:

- El tablero permite al usuario seleccionar entidades bancarias, años y otros parámetros para visualizar la tendencia de un indicador de riesgo financiero.
- Hay una sección para ingresar valores relacionados con la solvencia y otros indicadores financieros.
- Los gráficos se actualizan dinámicamente en función de las selecciones del usuario

Luego de entender que se hace en el código de la API y del tablero para su ejecución en una instancia de EC2 de AWS es necesario seguir los siguientes pasos.

1. Es necesario tener una carpeta raíz donde este todo el código desarrollado, estando en esa carpeta cree un repositorio git, y conéctelo con un repositorio remoto en GitHub (https://github.com/rmlopezgit/prec/).
2. Cree una instancia EC2 en AWS, en este caso se utilizó una instancia con imagen Ubuntu, t2micro, de 20 GB.
3. Instale todo lo necesario es su máquina virtual, ejecutador de Python3, librerías de Python, todo lo necesario.
4. Clone su repositorio remoto de GitHub en la máquina virtual, con git clone URL GitHub.
5. Por medio de los comandos ls y cd navegue hazla la carpeta de la API https://github.com/rmlopezgit/prec/tree/main/Code/app, y ejecute el archivo main.py con el comando python3 main.py.

6. En otro terminal por medio de los comandos ls y cd navegue hazla la carpeta del tablero https://github.com/rmlopezgit/prec/tree/main/Code, y ejecute el archivo Dash_2.py con el comando python3 Dash_2.py.
7. Podrá verificar que el API y el tablero están corriendo en la IP publica de la instancia de EC2 de AWS en los puertos 8001 y 8050 respectivamente.
8. El tablero se habrá instalado y puesto en productivo exitosamente.