Evaluación Inicial de Madurez de Capacidades en Ciencia de Datos

Naturgy – Asistencia Técnica sobre Framework Analítico IA/ML

17 febrero 2025 | Versión del documento 1.0

Contenido

1. Instrucciones generales 4

2. Evaluación 5

2.1. Bloque 1. Instrucciones Elementales 5

2.2. Bloque 2. Comprensión de datos 5

2.3. Bloque 3. Exploración y tratamiento de datos 5

2.4. Bloque 4. Modelización 6

Historial del documento

Versión	Fecha	Comentarios	Páginas afectadas
1.00	17/01/2025	Original	Todas

1. Instrucciones generales

- El dataset empleado para la siguiente prueba procede de un caso real anonimizado, de forma que no vas a saber que contienen las variables que estás manejando. Las únicas dos variables de las que conoces el contenido son:
    - *ID*: valor numérico que identifica el evento de forma unívoca.
    - *Target*: variable binaria que recoge el riesgo del evento, donde el 0 representa evento de no riesgo y el 1 de riesgo.
- El objetivo del ejercicio es evaluar la soltura en el manejo de Python para Ciencia de Datos.
- No importa el trasfondo del modelado, sólo se busca evaluar las capacidades de programación y compresión de resultados.
- No hay una única respuesta correcta. Se valorará tanto que la respuesta sea acertada para el planteamiento propuesto, como la justificación y explicaciones que se consideren.
- Se dispone de total libertad para instalar y usar las librerías de Python que consideres.
- Los resultados se han de entregar en un único fichero *.ipynb*
- El tiempo estimado para la resolución de esta prueba está estipulado en 3 horas.

2. Evaluación

2.1. Bloque 1. Instrucciones elementales

1. Muestra el directorio sobre el que se está trabajando. Cambia el directorio de trabajo a la carpeta que te han indicado al comienzo de la prueba.
2. Carga de paquetes. Verifica si están instalados los siguientes paquetes: Pandas, Seaborn, y Xlrd. Si no están, instálalos.
3. Carga las funciones contenidas en el *script wilcox1.py*
4. Establece una semilla para que los resultados sean replicables.
5. Crea un nuevo entorno virtual llamado "*venv".* No hace falta que instales nada en este entorno nuevo.

2.2. Bloque 2. Comprensión de datos

6. Realiza la carga del fichero *data_train.csv*. Las 373 deben cargarse como *string* y ser almacenadas en un *dataframe* denominado *raw_datos*.
7. ¿Cómo sería la sentencia si los datos proviniesen de un fichero en formato *.xlsx*?
8. Realiza un análisis descriptivo de todas las variables.
9. Localiza las variables con datos nulos (*NA, NaN*) y decide como tratar estas variables.
10. Recarga los datos como *src_train* dejando de Python decida el tipo más conveniente, a excepción de la variable ID que debe de ser de tipo string (*object*).
11. Sobre ese mismo *dataframe* elimina la primera variable.

2.3. Bloque 3. Exploración y tratamiento de datos

12. Grafica el histograma de la variable "var_5".
13. Realiza una tabla de frecuencias de cualquier variable del dataframe.
14. Realiza una tabla de frecuencias relativas de la variable Target.
15. Realiza un diagrama de cajas múltiple de la variable "var_2" por cada nivel de la variable "var_47".
16. Si existen *outliers* en el gráfico, imprímelos por pantalla.
17. De la variable "var_370" examina los *outliers*. Puedes utilizar la fórmula clásica Q1 -1.5*RI y Q3 + 1.5 * RI.
18. Sobre la variables "var_370", genera dos variables nuevas:

    a. Aplicando el logaritmo con base mediana.
    b. Normalizando dicha variable.

19. Construye una función que reciba una serie de datos (como puede ser una variable en un dataset) y devuelva esa serie normalizada entre 0 y 1. Además la función tiene que comprobar que el formato de la variable de entrada es una serie y elevar un error en caso contrario. Ejecuta esta función sobre la variable que consideres.
20. Genera los test unitarios que consideres para comprobar el funcionamiento correcto de la función anterior.
21. Genera un *dataframe* con las siguientes variables: 'ID', 'var_2', 'var_276', 'var_325', 'var_278', 'var_275', 'var_280', 'var_327', 'var_63', 'var_369', 'var_1', 'var_279', 'var_340', 'var_339', 'var_165', 'var_177', 'var_172', 'var_168', 'var_114', 'var_47', 'var_105', 'var_194', 'Target', además incluye las variables sintéticas generadas en la pregunta anterior. Guárdalo en un archivo *csv* que se llame *vun_train.csv*

2.4. Bloque 4. Modelización

22. Vuelve a cargar el archivo *vun_train.csv* sobre la variable *vun_train*.
23. Separa el *dataframe vun_train* en dos: uno de entrenamiento y otro de validación, con el 70% y el 30% de los registros respectivamente.
24. Realiza un modelo de regresión logística con las variables que consideres oportunas. Interpreta los resultados en función de los p-valores.
25. Realiza la matriz de confusión con los datos de validación. Analiza los resultados con la métrica que consideres más oportuna.