Team Challenge: Lost in ML

Herramientas básicas para ML

THE BRIDGE

Objetivo del negocio

Objetivo: Desarrollar un conjunto de herramientas fundamentales que simplifiquen y automaticen análisis repetitivos en EDA para la selección de features en modelos de Machine Learning (aprendizaje automático).

Problema de negocio: Reducir significativamente el tiempo y los costos computacionales asociados al análisis, mejorando el flujo de trabajo en proyectos de Machine Learning.

Archivos

DataFrame:

El dataset utilizado para el ejemplo de Machine Learning es:
data/student_performance_data.csv
Fuente: https://www.kaggle.com/datasets/rabieelkharoua/students-performance-dataset

toolbox_ML.py:

Este conjunto de código incluye las funciones fundamentales necesarias para el preprocesamiento de datos y la exploración de conjuntos de datos para la selección de features.

Lost_in_ML_ToolBox.ipynb:

Notebook con pruebas de funciones y corrección de errores.

THE BRIDGE

1. describe_df()

Análisis descriptivo del DataFrame.

	Age	Gender	Ethnicity	ParentalEducation	StudyTimeWeekly
DATA_TYPE	int64	int64	int64	int64	float64
MISSINGS(%)	0.0	0.0	0.0	0.0	0.0
UNIQUE_VALUES	4	2	4	5	2392
CARDIN(%)	0.167	0.084	0.167	0.209	100.0

..

.

1. describe_df()

Testeo de errores: describe_df(155)

El error se debe a que se ha pasado un argumento incorrecto (un entero = 155), lo que provoca el error.

```
[9]  describe_df(155)

...  ---------------------------------------------------------------------------
     ValueError                                Traceback (most recent call last)
     Cell In[9], line 1
     ----> 1 describe_df(155)

     File c:\Users\lucas\Documents\Cursos\The_Bridge\BOOTCAMP\Online_env\Team-Challenge---Lost-in-ML\toolbox_ML.py:33, in describe_df(df)
          21 """
          22 Función para describir un DataFrame de pandas proporcionando información sobre el tipo de datos,
          23 valores faltantes, valores únicos y cardinalidad.
        (...)
          29     DataFrame con la información recopilada sobre el DataFrame de entrada.
          30 """
          32 if not isinstance(df, pd.DataFrame):
     ---> 33     raise ValueError("El argumento 'df' debe ser un DataFrame de pandas válido.")
          35 # Creamons un diccionario para almacenar la información
          36 data = {
          37     'DATA_TYPE': df.dtypes,
          38     'MISSINGS(%)': df.isnull().mean() * 100,
          39     'UNIQUE_VALUES': df.nunique(),
          40     'CARDIN(%)': round(df.nunique() / len(df) * 100, 3)
          41 }

     ValueError: El argumento 'df' debe ser un DataFrame de pandas válido.
```

THE BRIDGE

2. tipifica_variable()

	nombre_variable	tipo_sugerido	tipo_real
0	Age	Categórica	int64
1	Gender	Binaria	int64
2	Ethnicity	Categórica	int64
3	ParentalEducation	Numerica Discreta	int64
4	StudyTimeWeekly	Numerica Continua	float64
5	Absences	Numerica Discreta	int64
6	Tutoring	Binaria	int64
7	ParentalSupport	Numerica Discreta	int64
8	Extracurricular	Binaria	int64
9	Sports	Binaria	int64
10	Music	Binaria	int64
11	Volunteering	Binaria	int64
12	GPA	Numerica Continua	float64
13	GradeClass	Numerica Discreta	float64

THE BRIDGE

3. get_features_num_regression()

```
1  # Testeo con correlación de 0.2
2  get_features_num_regression(df_estudiantes, "GPA", 0.2)
```

```
['Absences', 'GradeClass']
```

Testeo de errores:

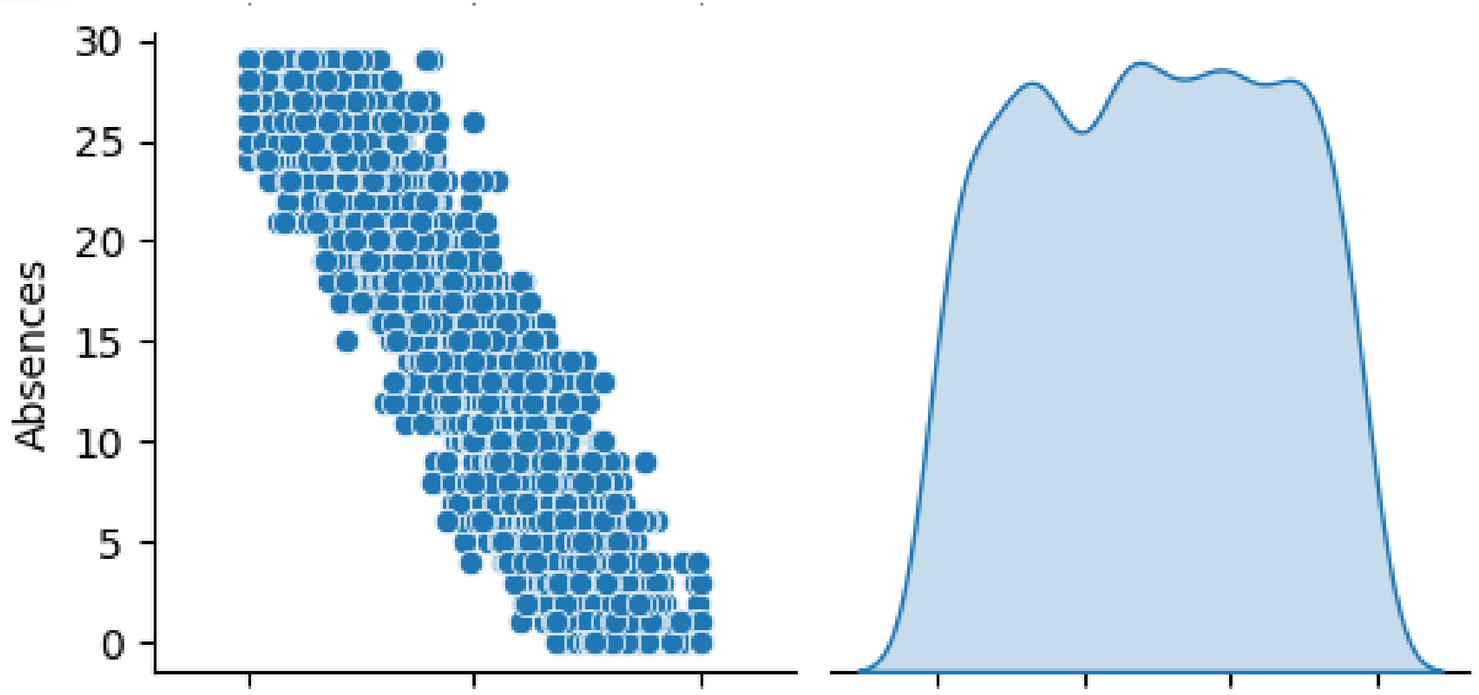
```
1  get_features_num_regression("df_estudiantes", "GPA", 0.7, pvalue=0.01)
```

```
El primer argumento debe ser un DataFrame.
```

THE BRIDGE

4. plot_features_num_regression()

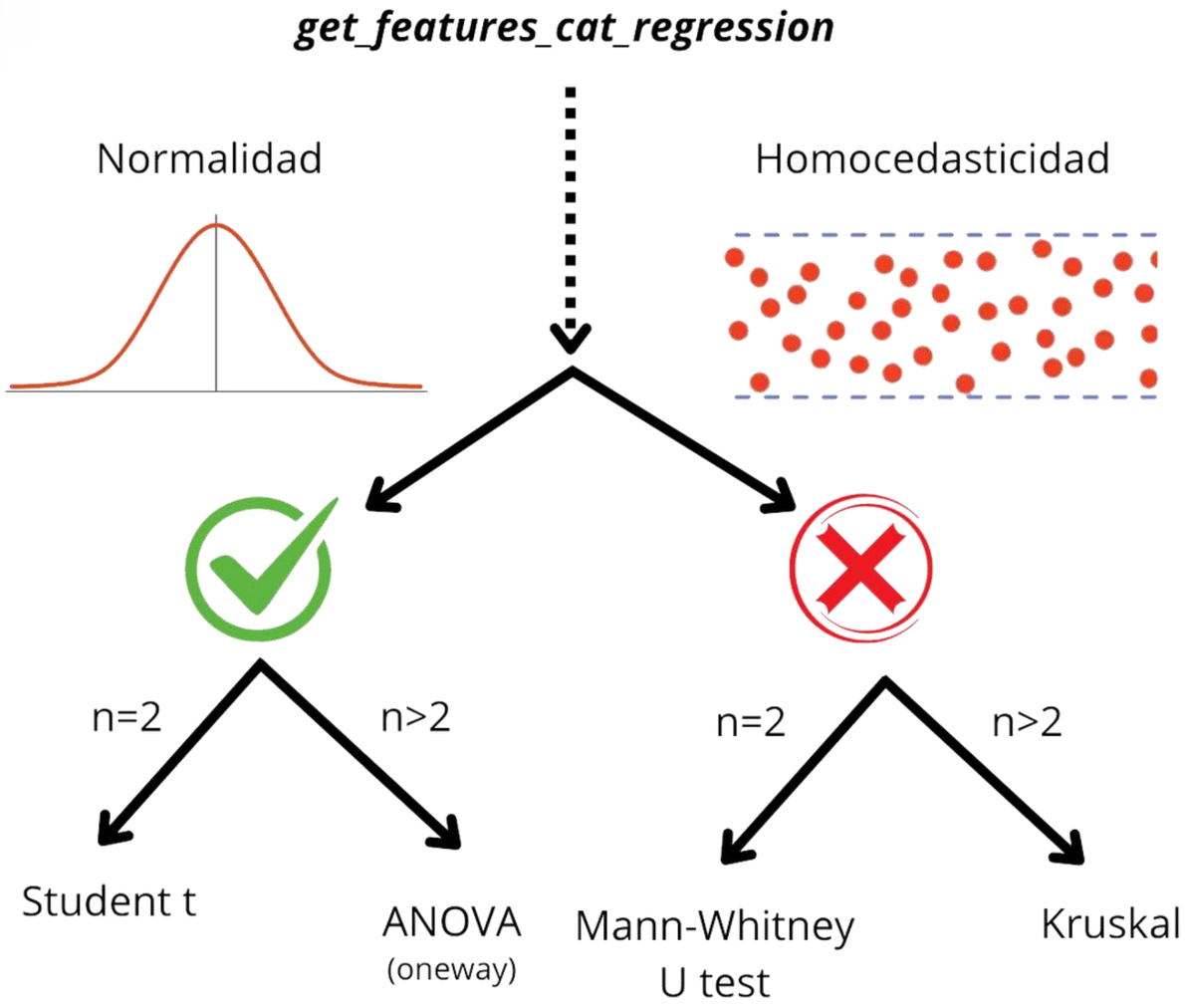
```
plot_features_num_regression(df_estudiantes, target_col="GPA", columns=[], umbral_corr=0.7, pvalue=0.01)
```




```
['Absences', 'GradeClass']
```

THE BRIDGE

5. get_features_cat_regression()



THE BRIDGE

5. get_features_cat_regression()

```
1  get_features_cat_regression(df_teste_categoricas, "GPA")
2
```

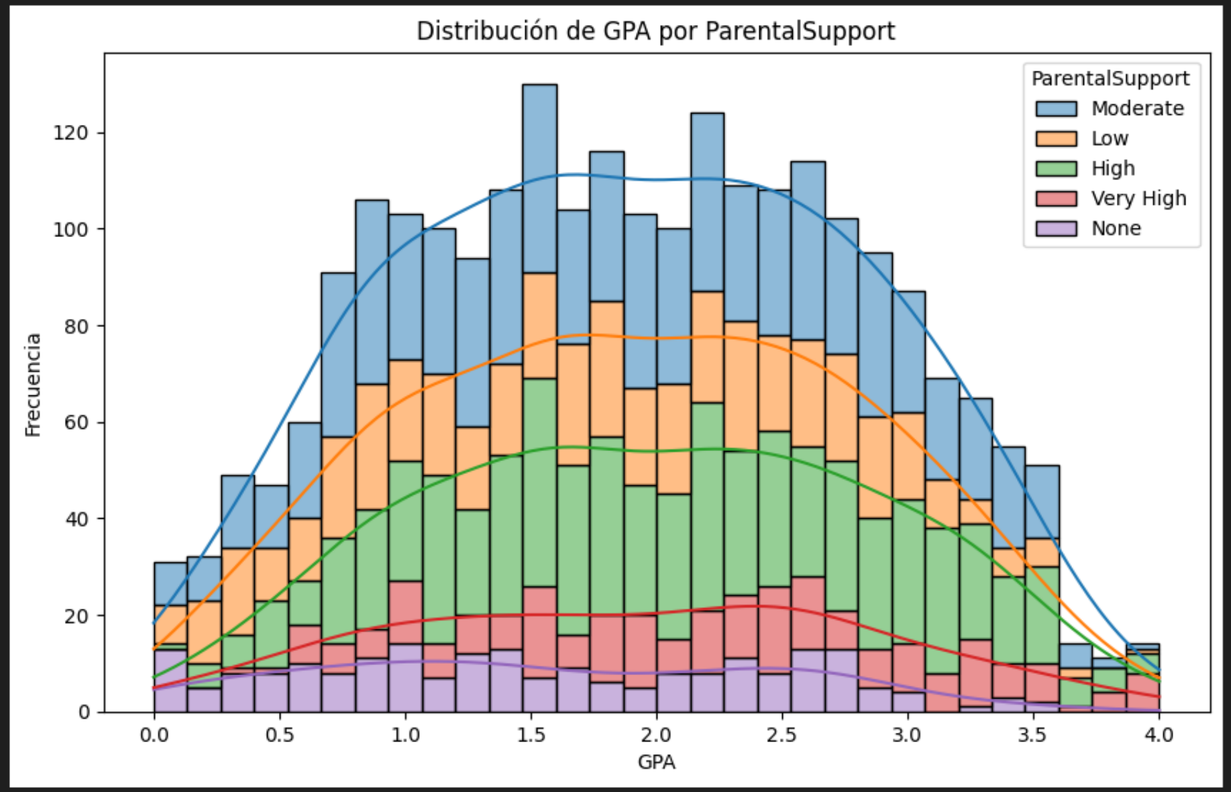
```
La distribución de GPA NO es normal o homocedástica.
Kruskal (p_value): 8.140314082661161e-16

['ParentalSupport']
```

THE BRIDGE

6. plot_features_cat_regression()

Testeo de la función ***plot_features_cat_regression()***

```
[53] plot_features_cat_regression(df_teste_categoricas, target_col="GPA", columns=[], pvalue=0.05, with_individual_plot=False)
```




```
['ParentalSupport']
```

THE BRIDGE