

✓ 03.1_Division_TrainValTest

Objetivo

Cargar el dataset limpio de la encuesta NFCS 2021, explorar rápidamente su estructura, derivar algunas variables básicas y dividir el conjunto de datos en particiones estratificadas de entrenamiento, validación y prueba para su posterior modelado.

Entradas (Inputs)

- `data/processed/final/NFCS_2021_final_clean.csv`

Salidas (Outputs)

- `data/splits/final/X_train.parquet`
- `data/splits/final/y_train.parquet`
- `data/splits/final/X_val.parquet`
- `data/splits/final/y_val.parquet`
- `data/splits/final/X_test.parquet`
- `data/splits/final/y_test.parquet`

+ Código

+ Texto

Resumen Ejecutivo

- El notebook aborda la preparación de los datos limpios de la encuesta NFCS 2021 para un proyecto de modelado de riesgo financiero.
- Se importa el dataset desde Google Drive y se definen las librerías clave (`pandas`, `NumPy`, `scikit-learn`, `pyarrow`).
- Se crea la variable **FL_SCORE** (suma de aciertos en preguntas de alfabetización financiera) y una métrica de **PORTFOLIO_DIVERSITY**.
- Se especifican las columnas de identificación (`NFCSID`), el objetivo (`B10`) y el peso de muestreo (`WGT1`).
- Se construye la matriz de características **X** y el vector objetivo **y**, eliminando IDs y columnas no predictivas.
- Se realiza una división estratificada en tres conjuntos: entrenamiento ($\approx 70\%$), validación ($\approx 15\%$) y test ($\approx 15\%$) usando `train_test_split`.
- Los tamaños resultantes son **X_train** (1976×92), **X_val** (424×92) y **X_test** (392×92), manteniendo la distribución de la variable objetivo.
- Finalmente, se guardan los tres “splits” en formato Parquet mediante `pyarrow` para facilitar su carga en etapas posteriores.

✓ 1. Carga de dependencias y montaje de Google Drive

Agrupa e importa todas las librerías necesarias (estándar, terceros y locales) y monta Google Drive para acceder a los datos.

```
import sys
from pathlib import Path

from google.colab import drive
import numpy as np
import pandas as pd
from sklearn.model_selection import train_test_split

# Configuración de semilla para reproducibilidad
RANDOM_STATE = 42

# Montar Google Drive
drive.mount('/content/drive', force_remount=True)

# Añadir la raíz del proyecto al path para poder importar 'config'
ROOT_PATH_STR = '/content/drive/MyDrive/TFM-AntonioEsquinas'
if ROOT_PATH_STR not in sys.path:
    sys.path.append(ROOT_PATH_STR)

# Importar las rutas necesarias desde el archivo de configuración
# Se importa la ruta del dataset procesado final y la de los splits finales.
from config import FINAL_PROCESSED_DATA_DIR, FINAL_SPLITS_DIR, METADATA_DIR

# Definir la ruta del archivo de entrada usando las variables de config
DATA_FILE_PATH = FINAL_PROCESSED_DATA_DIR / 'NFCS_2021_final_clean.csv'

# Leer el dataset
df = pd.read_csv(DATA_FILE_PATH)
print(f"Dataset cargado con {df.shape[0]} filas y {df.shape[1]} columnas")

print('\n Primeras filas del dataset:')
display(df.head())

print('\n Información general:')
display(df.info())

print('\n Estadísticas descriptivas:')
display(df.describe())

# Conteo de valores nulos
TARGET = 'B10' # Asegúrate de que esta es tu variable objetivo
na_pct_full = df.isna().mean().sort_values(ascending=False)
print('\n Top 10 variables con más % de NA:')
display(na_pct_full.head(10))

# Distribución de la variable objetivo
```

```
if TARGET in df.columns:  
    print(f"\n Distribución de la variable objetivo '{TARGET}':")  
    display(df[TARGET].value_counts(normalize=True))  
else:  
    print(f"\n La columna {TARGET} no existe en el dataset.")
```

Mounted at /content/drive

Módulo de configuración cargado y estructura de carpetas asegurada.
Dataset cargado con 2824 filas y 92 columnas

Primeras filas del dataset:

	NFCSID	A1	A2	A3	B2_1	B2_2	B2_3	B2_4	B2_5	B2_20	...	G12	G13	H31
0	2.021010e+09	1.0	1.0	1.0	1.0	2.0	1.0	2.0	2.0	1.0	...	2.0	3.0	1.0
1	2.021010e+09	1.0	1.0	1.0	1.0	2.0	1.0	2.0	2.0	1.0	...	3.0	3.0	1.0
2	2.021010e+09	1.0	1.0	1.0	1.0	1.0	1.0	2.0	2.0	2.0	...	2.0	4.0	2.0
3	2.021010e+09	2.0	2.0	1.0	2.0	2.0	1.0	2.0	2.0	2.0	...	2.0	3.0	2.0
4	2.021010e+09	1.0	1.0	1.0	1.0	2.0	1.0	2.0	2.0	2.0	...	3.0	2.0	2.0

5 rows × 92 columns

Información general:

<class 'pandas.core.frame.DataFrame'>

RangeIndex: 2824 entries, 0 to 2823

Data columns (total 92 columns):

#	Column	Non-Null Count	Dtype
0	NFCSID	2824 non-null	float64
1	A1	2824 non-null	float64
2	A2	2824 non-null	float64
3	A3	2824 non-null	float64
4	B2_1	2824 non-null	float64
5	B2_2	2824 non-null	float64
6	B2_3	2824 non-null	float64
7	B2_4	2824 non-null	float64
8	B2_5	2824 non-null	float64
9	B2_20	2824 non-null	float64
10	B2_23	2824 non-null	float64
11	B2_24	2824 non-null	float64
12	B30	2824 non-null	float64
13	B31	2824 non-null	float64
14	B3	2824 non-null	float64
15	B32	2824 non-null	float64
16	B4	2824 non-null	float64
17	B10	2824 non-null	float64
18	B11	2824 non-null	float64
19	B35	2824 non-null	float64
20	B23	2824 non-null	float64
21	B24	2824 non-null	float64
22	B25	2824 non-null	float64
23	B26	2824 non-null	float64
24	C22_1	2824 non-null	float64
25	C22_2	2824 non-null	float64
26	C22_3	2824 non-null	float64
27	C22_4	2824 non-null	float64
28	C24	2824 non-null	float64
29	C25	2824 non-null	float64
30	C26	2824 non-null	float64
31	C30	2824 non-null	float64
32	C7	2824 non-null	float64
33	D1_1	2824 non-null	float64
34	D1_2	2824 non-null	float64
35	...	2824 non-null	float64

```
35  B2      2824 non-null   float64
36  B3      2824 non-null   float64
37  D21     2824 non-null   float64
38  D30     2824 non-null   float64
El resultado muestra la distribución de la variable proportion en el DataFrame inicial, donde la clase 3.0 representa el 54.99 % de los datos, seguida de la clase 2.0 con el 27.34 %, la clase 4.0 con el 9.56 % y la clase 12.71 con el 8.11 %. Este desbalance de clases es clave a la hora de estratificar los datos o aplicar técnicas de balanceo para evitar sesgos en el modelo.
```

2. Definición de la variable FL_SCORE

Define el nombre de la variable que almacenará la puntuación de alfabetización financiera.

```
41  E1_1    2824 non-null   float64
42  E5      2824 non-null   float64
43  F30_1   2824 non-null   float64
44  F30_2   2824 non-null   float64
45  F30_3   2824 non-null   float64
46  F30_4   2824 non-null   float64
47  F30_5   2824 non-null   float64
48  F30_6   2824 non-null   float64
49  F30_7   2824 non-null   float64
```

```
# FL_SCORE = suma de aciertos en preguntas de alfabetización financiera
literacy_items = ['G4', 'G5', 'G6', 'G7', 'G8', 'G11', 'G12', 'G13', 'G21']
literacy_present = [col for col in literacy_items if col in df.columns]
if literacy_present:
    df['FL_SCORE'] = df[literacy_present].sum(axis=1)
    print("Variable FL_SCORE creada correctamente.")
else:
    print("Atención: Ninguno de los literacy_items está en df. No se creó FL_SCORE.")
```

```
# PORTFOLIO_DIVERSITY = número de activos marcados (B2_)
asset_cols = [c for c in df.columns if c.startswith('B2_')]
if asset_cols:
    df['PORTFOLIO_DIVERSITY'] = (df[asset_cols] == 1).sum(axis=1)
    print("Variable PORTFOLIO_DIVERSITY creada correctamente.")
else:
    print("Atención: No se encontraron columnas B2_. No se creó PORTFOLIO_DIVERSITY.")
```

```
# TRADER_SCORE = inversamente proporcional a B3 + ajuste por B31
if 'B3' in df.columns and 'B31' in df.columns:
    B3_norm = df['B3'].replace({98: np.nan, 99: np.nan})
    df['TRADER_SCORE'] = (5 - B3_norm).add(df['B31'].map({1:1, 2:0}), fill_value=0)
    print("Variable TRADER_SCORE creada correctamente.")
else:
    print("Atención: Las columnas B3 y/o B31 no existen. No se creó TRADER_SCORE.")
```

```
# **Verificamos que efectivamente se hayan agregado** las nuevas columnas a df
print("Columnas actuales de df:")
print(df.columns.tolist())
print("Total columnas en df:", len(df.columns))
print("Dimensiones tras derivar variables:", df.shape)
```

```
50  G15      2824 non-null   float64
51  Variable FL_SCORE 2824 non-null   float64
52  Variable PORTFOLIO_DIVERSITY 2824 non-null   float64
53  Variable TRADER_SCORE 2824 non-null   float64
54  Columnas actuales de df:
55  ['B2_1', 'B2_2', 'B2_3', 'B2_4', 'B2_5', 'B2_6', 'B2_7', 'B2_8', 'B2_9', 'B2_10', 'B2_11', 'B2_12', 'B2_13', 'B2_14', 'B2_15', 'B2_16', 'B2_17', 'B2_18', 'B2_19', 'B2_20', 'B2_21', 'B2_22', 'B2_23', 'B2_24', 'B2_25', 'B2_26', 'B2_27', 'B2_28', 'B2_29', 'B2_30', 'B2_31', 'B2_32', 'B2_33', 'B2_34', 'B2_35', 'B2_36', 'B2_37', 'B2_38', 'B2_39', 'B2_40', 'B2_41', 'B2_42', 'B2_43', 'B2_44', 'B2_45', 'B2_46', 'B2_47', 'B2_48', 'B2_49', 'B2_50', 'B2_51', 'B2_52', 'B2_53', 'B2_54', 'B2_55', 'B2_56', 'B2_57', 'B2_58', 'B2_59', 'B2_60', 'B2_61', 'B2_62', 'B2_63', 'B2_64', 'B2_65', 'B2_66', 'B2_67', 'B2_68', 'B2_69', 'B2_70', 'B2_71', 'B2_72', 'B2_73', 'B2_74', 'B2_75', 'B2_76', 'B2_77', 'B2_78', 'B2_79', 'B2_80', 'B2_81', 'B2_82', 'B2_83', 'B2_84', 'B2_85', 'B2_86', 'B2_87', 'B2_88', 'B2_89', 'B2_90', 'B2_91', 'B2_92', 'B2_93', 'B2_94', 'B2_95', 'B2_96', 'B2_97', 'B2_98', 'B2_99', 'B2_100', 'B2_101', 'B2_102', 'B2_103', 'B2_104', 'B2_105', 'B2_106', 'B2_107', 'B2_108', 'B2_109', 'B2_110', 'B2_111', 'B2_112', 'B2_113', 'B2_114', 'B2_115', 'B2_116', 'B2_117', 'B2_118', 'B2_119', 'B2_120', 'B2_121', 'B2_122', 'B2_123', 'B2_124', 'B2_125', 'B2_126', 'B2_127', 'B2_128', 'B2_129', 'B2_130', 'B2_131', 'B2_132', 'B2_133', 'B2_134', 'B2_135', 'B2_136', 'B2_137', 'B2_138', 'B2_139', 'B2_140', 'B2_141', 'B2_142', 'B2_143', 'B2_144', 'B2_145', 'B2_146', 'B2_147', 'B2_148', 'B2_149', 'B2_150', 'B2_151', 'B2_152', 'B2_153', 'B2_154', 'B2_155', 'B2_156', 'B2_157', 'B2_158', 'B2_159', 'B2_160', 'B2_161', 'B2_162', 'B2_163', 'B2_164', 'B2_165', 'B2_166', 'B2_167', 'B2_168', 'B2_169', 'B2_170', 'B2_171', 'B2_172', 'B2_173', 'B2_174', 'B2_175', 'B2_176', 'B2_177', 'B2_178', 'B2_179', 'B2_180', 'B2_181', 'B2_182', 'B2_183', 'B2_184', 'B2_185', 'B2_186', 'B2_187', 'B2_188', 'B2_189', 'B2_190', 'B2_191', 'B2_192', 'B2_193', 'B2_194', 'B2_195', 'B2_196', 'B2_197', 'B2_198', 'B2_199', 'B2_200', 'B2_201', 'B2_202', 'B2_203', 'B2_204', 'B2_205', 'B2_206', 'B2_207', 'B2_208', 'B2_209', 'B2_210', 'B2_211', 'B2_212', 'B2_213', 'B2_214', 'B2_215', 'B2_216', 'B2_217', 'B2_218', 'B2_219', 'B2_220', 'B2_221', 'B2_222', 'B2_223', 'B2_224', 'B2_225', 'B2_226', 'B2_227', 'B2_228', 'B2_229', 'B2_230', 'B2_231', 'B2_232', 'B2_233', 'B2_234', 'B2_235', 'B2_236', 'B2_237', 'B2_238', 'B2_239', 'B2_240', 'B2_241', 'B2_242', 'B2_243', 'B2_244', 'B2_245', 'B2_246', 'B2_247', 'B2_248', 'B2_249', 'B2_250', 'B2_251', 'B2_252', 'B2_253', 'B2_254', 'B2_255', 'B2_256', 'B2_257', 'B2_258', 'B2_259', 'B2_260', 'B2_261', 'B2_262', 'B2_263', 'B2_264', 'B2_265', 'B2_266', 'B2_267', 'B2_268', 'B2_269', 'B2_270', 'B2_271', 'B2_272', 'B2_273', 'B2_274', 'B2_275', 'B2_276', 'B2_277', 'B2_278', 'B2_279', 'B2_280', 'B2_281', 'B2_282', 'B2_283', 'B2_284', 'B2_285', 'B2_286', 'B2_287', 'B2_288', 'B2_289', 'B2_290', 'B2_291', 'B2_292', 'B2_293', 'B2_294', 'B2_295', 'B2_296', 'B2_297', 'B2_298', 'B2_299', 'B2_300', 'B2_301', 'B2_302', 'B2_303', 'B2_304', 'B2_305', 'B2_306', 'B2_307', 'B2_308', 'B2_309', 'B2_310', 'B2_311', 'B2_312', 'B2_313', 'B2_314', 'B2_315', 'B2_316', 'B2_317', 'B2_318', 'B2_319', 'B2_320', 'B2_321', 'B2_322', 'B2_323', 'B2_324', 'B2_325', 'B2_326', 'B2_327', 'B2_328', 'B2_329', 'B2_330', 'B2_331', 'B2_332', 'B2_333', 'B2_334', 'B2_335', 'B2_336', 'B2_337', 'B2_338', 'B2_339', 'B2_340', 'B2_341', 'B2_342', 'B2_343', 'B2_344', 'B2_345', 'B2_346', 'B2_347', 'B2_348', 'B2_349', 'B2_350', 'B2_351', 'B2_352', 'B2_353', 'B2_354', 'B2_355', 'B2_356', 'B2_357', 'B2_358', 'B2_359', 'B2_360', 'B2_361', 'B2_362', 'B2_363', 'B2_364', 'B2_365', 'B2_366', 'B2_367', 'B2_368', 'B2_369', 'B2_370', 'B2_371', 'B2_372', 'B2_373', 'B2_374', 'B2_375', 'B2_376', 'B2_377', 'B2_378', 'B2_379', 'B2_380', 'B2_381', 'B2_382', 'B2_383', 'B2_384', 'B2_385', 'B2_386', 'B2_387', 'B2_388', 'B2_389', 'B2_390', 'B2_391', 'B2_392', 'B2_393', 'B2_394', 'B2_395', 'B2_396', 'B2_397', 'B2_398', 'B2_399', 'B2_400', 'B2_401', 'B2_402', 'B2_403', 'B2_404', 'B2_405', 'B2_406', 'B2_407', 'B2_408', 'B2_409', 'B2_410', 'B2_411', 'B2_412', 'B2_413', 'B2_414', 'B2_415', 'B2_416', 'B2_417', 'B2_418', 'B2_419', 'B2_420', 'B2_421', 'B2_422', 'B2_423', 'B2_424', 'B2_425', 'B2_426', 'B2_427', 'B2_428', 'B2_429', 'B2_430', 'B2_431', 'B2_432', 'B2_433', 'B2_434', 'B2_435', 'B2_436', 'B2_437', 'B2_438', 'B2_439', 'B2_440', 'B2_441', 'B2_442', 'B2_443', 'B2_444', 'B2_445', 'B2_446', 'B2_447', 'B2_448', 'B2_449', 'B2_450', 'B2_451', 'B2_452', 'B2_453', 'B2_454', 'B2_455', 'B2_456', 'B2_457', 'B2_458', 'B2_459', 'B2_460', 'B2_461', 'B2_462', 'B2_463', 'B2_464', 'B2_465', 'B2_466', 'B2_467', 'B2_468', 'B2_469', 'B2_470', 'B2_471', 'B2_472', 'B2_473', 'B2_474', 'B2_475', 'B2_476', 'B2_477', 'B2_478', 'B2_479', 'B2_480', 'B2_481', 'B2_482', 'B2_483', 'B2_484', 'B2_485', 'B2_486', 'B2_487', 'B2_488', 'B2_489', 'B2_490', 'B2_491', 'B2_492', 'B2_493', 'B2_494', 'B2_495', 'B2_496', 'B2_497', 'B2_498', 'B2_499', 'B2_500', 'B2_501', 'B2_502', 'B2_503', 'B2_504', 'B2_505', 'B2_506', 'B2_507', 'B2_508', 'B2_509', 'B2_510', 'B2_511', 'B2_512', 'B2_513', 'B2_514', 'B2_515', 'B2_516', 'B2_517', 'B2_518', 'B2_519', 'B2_520', 'B2_521', 'B2_522', 'B2_523', 'B2_524', 'B2_525', 'B2_526', 'B2_527', 'B2_528', 'B2_529', 'B2_530', 'B2_531', 'B2_532', 'B2_533', 'B2_534', 'B2_535', 'B2_536', 'B2_537', 'B2_538', 'B2_539', 'B2_540', 'B2_541', 'B2_542', 'B2_543', 'B2_544', 'B2_545', 'B2_546', 'B2_547', 'B2_548', 'B2_549', 'B2_550', 'B2_551', 'B2_552', 'B2_553', 'B2_554', 'B2_555', 'B2_556', 'B2_557', 'B2_558', 'B2_559', 'B2_560', 'B2_561', 'B2_562', 'B2_563', 'B2_564', 'B2_565', 'B2_566', 'B2_567', 'B2_568', 'B2_569', 'B2_570', 'B2_571', 'B2_572', 'B2_573', 'B2_574', 'B2_575', 'B2_576', 'B2_577', 'B2_578', 'B2_579', 'B2_580', 'B2_581', 'B2_582', 'B2_583', 'B2_584', 'B2_585', 'B2_586', 'B2_587', 'B2_588', 'B2_589', 'B2_590', 'B2_591', 'B2_592', 'B2_593', 'B2_594', 'B2_595', 'B2_596', 'B2_597', 'B2_598', 'B2_599', 'B2_600', 'B2_601', 'B2_602', 'B2_603', 'B2_604', 'B2_605', 'B2_606', 'B2_607', 'B2_608', 'B2_609', 'B2_610', 'B2_611', 'B2_612', 'B2_613', 'B2_614', 'B2_615', 'B2_616', 'B2_617', 'B2_618', 'B2_619', 'B2_620', 'B2_621', 'B2_622', 'B2_623', 'B2_624', 'B2_625', 'B2_626', 'B2_627', 'B2_628', 'B2_629', 'B2_630', 'B2_631', 'B2_632', 'B2_633', 'B2_634', 'B2_635', 'B2_636', 'B2_637', 'B2_638', 'B2_639', 'B2_640', 'B2_641', 'B2_642', 'B2_643', 'B2_644', 'B2_645', 'B2_646', 'B2_647', 'B2_648', 'B2_649', 'B2_650', 'B2_651', 'B2_652', 'B2_653', 'B2_654', 'B2_655', 'B2_656', 'B2_657', 'B2_658', 'B2_659', 'B2_660', 'B2_661', 'B2_662', 'B2_663', 'B2_664', 'B2_665', 'B2_666', 'B2_667', 'B2_668', 'B2_669', 'B2_670', 'B2_671', 'B2_672', 'B2_673', 'B2_674', 'B2_675', 'B2_676', 'B2_677', 'B2_678', 'B2_679', 'B2_680', 'B2_681', 'B2_682', 'B2_683', 'B2_684', 'B2_685', 'B2_686', 'B2_687', 'B2_688', 'B2_689', 'B2_690', 'B2_691', 'B2_692', 'B2_693', 'B2_694', 'B2_695', 'B2_696', 'B2_697', 'B2_698', 'B2_699', 'B2_700', 'B2_701', 'B2_702', 'B2_703', 'B2_704', 'B2_705', 'B2_706', 'B2_707', 'B2_708', 'B2_709', 'B2_710', 'B2_711', 'B2_712', 'B2_713', 'B2_714', 'B2_715', 'B2_716', 'B2_717', 'B2_718', 'B2_719', 'B2_720', 'B2_721', 'B2_722', 'B2_723', 'B2_724', 'B2_725', 'B2_726', 'B2_727', 'B2_728', 'B2_729', 'B2_730', 'B2_731', 'B2_732', 'B2_733', 'B2_734', 'B2_735', 'B2_736', 'B2_737', 'B2_738', 'B2_739', 'B2_740', 'B2_741', 'B2_742', 'B2_743', 'B2_744', 'B2_745', 'B2_746', 'B2_747', 'B2_748', 'B2_749', 'B2_750', 'B2_751', 'B2_752', 'B2_753', 'B2_754', 'B2_755', 'B2_756', 'B2_757', 'B2_758', 'B2_759', 'B2_760', 'B2_761', 'B2_762', 'B2_763', 'B2_764', 'B2_765', 'B2_766', 'B2_767', 'B2_768', 'B2_769', 'B2_770', 'B2_771', 'B2_772', 'B2_773', 'B2_774', 'B2_775', 'B2_776', 'B2_777', 'B2_778', 'B2_779', 'B2_780', 'B2_781', 'B2_782', 'B2_783', 'B2_784', 'B2_785', 'B2_786', 'B2_787', 'B2_788', 'B2_789', 'B2_790', 'B2_791', 'B2_792', 'B2_793', 'B2_794', 'B2_795', 'B2_796', 'B2_797', 'B2_798', 'B2_799', 'B2_800', 'B2_801', 'B2_802', 'B2_803', 'B2_804', 'B2_805', 'B2_806', 'B2_807', 'B2_808', 'B2_809', 'B2_810', 'B2_811', 'B2_812', 'B2_813', 'B2_814', 'B2_815', 'B2_816', 'B2_817', 'B2_818', 'B2_819', 'B2_820', 'B2_821', 'B2_822', 'B2_823', 'B2_824', 'B2_825', 'B2_826', 'B2_827', 'B2_828', 'B2_829', 'B2_830', 'B2_831', 'B2_832', 'B2_833', 'B2_834', 'B2_835', 'B2_836', 'B2_837', 'B2_838', 'B2_839', 'B2_840', 'B2_841', 'B2_842', 'B2_843', 'B2_844', 'B2_845', 'B2_846', 'B2_847', 'B2_848', 'B2_849', 'B2_850', 'B2_851', 'B2_852', 'B2_853', 'B2_854', 'B2_855', 'B2_856', 'B2_857', 'B2_858', 'B2_859', 'B2_860', 'B2_861', 'B2_862', 'B2_863', 'B2_864', 'B2_865', 'B2_866', 'B2_867', 'B2_868', 'B2_869', 'B2_870', 'B2_871', 'B2_872', 'B2_873', 'B2_874', 'B2_875', 'B2_876', 'B2_877', 'B2_878', 'B2_879', 'B2_880', 'B2_881', 'B2_882', 'B2_883', 'B2_884', 'B2_885', 'B2_886', 'B2_887', 'B2_888', 'B2_889', 'B2_890', 'B2_891', 'B2_892', 'B2_893', 'B2_894', 'B2_895', 'B2_896', 'B2_897', 'B2_898', 'B2_899', 'B2_900', 'B2_901', 'B2_902', 'B2_903', 'B2_904', 'B2_905', 'B2_906', 'B2_907', 'B2_908', 'B2_909', 'B2_910', 'B2_911', 'B2_912', 'B2_913', 'B2_914', 'B2_915', 'B2_916', 'B2_917', 'B2_918', 'B2_919', 'B2_920', 'B2_921', 'B2_922', 'B2_923', 'B2_924', 'B2_925', 'B2_926', 'B2_927', 'B2_928', 'B2_929', 'B2_930', 'B2_931', 'B2_932', 'B2_933', 'B2_934', 'B2_935', 'B2_936', 'B2_937', 'B2_938', 'B2_939', 'B2_940', 'B2_941', 'B2_942', 'B2_943', 'B2_944', 'B2_945', 'B2_946', 'B2_947', 'B2_948', 'B2_949', 'B2_950', 'B2_951', 'B2_952', 'B2_953', 'B2_954', 'B2_955', 'B2_956', 'B2_957', 'B2_958', 'B2_959', 'B2_960', 'B2_961', 'B2_962', 'B2_963', 'B2_964', 'B2_965', 'B2_966', 'B2_967', 'B2_968', 'B2_969', 'B2_970', 'B2_971', 'B2_972', 'B2_973', 'B2_974', 'B2_975', 'B2_976', 'B2_977', 'B2_978', 'B2_979', 'B2_980', 'B2_981', 'B2_982', 'B2_983', 'B2_984', 'B2_985', 'B2_986', 'B2_987', 'B2_988', 'B2_989', 'B2_990', 'B2_991', 'B2_992', 'B2_993', 'B2_994', 'B2_995', 'B2_996', 'B2_997', 'B2_998', 'B2_999', 'B2_1000', 'B2_1001', 'B2_1002', 'B2_1003', 'B2_1004', 'B2_1005', 'B2_1006', 'B2_1007', 'B2_1008', 'B2_1009', 'B2_1010', 'B2_1011', 'B2_1012', 'B2_1013', 'B2_1014', 'B2_1015', 'B2_1016', 'B2_1017', 'B2_1018', 'B2_1019', 'B2_1020', 'B2_1021', 'B2_1022', 'B2_1023', 'B2_1024', 'B2_1025', 'B2_1026', 'B2_1027', 'B2_1028', 'B2_1029', 'B2_1030', 'B2_1031', 'B2_1032', 'B2_1033', 'B2_1034', 'B2_1035', 'B2_1036', 'B2_1037', 'B2_1038', 'B2_1039', 'B2_1040', 'B2_1041', 'B2_1042', 'B2_1043', 'B2_1044', 'B2_1045', 'B2_1046', 'B2_1047', 'B2_1048', 'B2_1049', 'B2_1050', 'B2_1051', 'B2_1052', 'B2_1053', 'B2_1054', 'B2_1055', 'B2_1056', 'B2_1057', 'B2_1058', 'B2_1059', 'B2_1060', 'B2_1061', 'B2_1062', 'B2_1063', 'B2_1064', 'B2_1065', 'B2_1066', 'B2_1067', 'B2_1068', 'B2_1069', 'B2_1070', 'B2_1071', 'B2_1072', 'B2_1073', 'B2_1074', 'B2_1075', 'B2_1076', 'B2_1077', 'B2_1078', 'B2_1079', 'B2_1080', 'B2_1081', 'B2_1082', 'B2_1083', 'B2_1084', 'B2_1085', 'B2_1086', 'B2_1087', 'B2_1088', 'B2_1089', 'B2_1090', 'B2_1091', 'B2_1092', 'B2_1093', 'B2_1094', 'B2_1095', 'B2_1096', 'B2_1097', 'B2_1098', 'B2_1099', 'B2_1100', 'B2_1101', 'B2_1102', 'B2_1103', 'B2_1104', 'B2_1105', 'B2_1106', 'B2_1107', 'B2_1108', 'B2_1109', 'B2_1110', 'B2_1111', 'B2_1112', 'B2_1113', 'B2_1114', 'B2_1115', 'B2_1116', 'B2_1117', 'B2_1118', 'B2_1119', 'B2_1120', 'B2_1121', 'B2_1122', 'B2_1123', 'B2_1124', 'B2_1125', 'B2_1126', 'B2_1127', 'B2_1128', 'B2_1129', 'B2_1130', 'B2_1131', 'B2_1132', 'B2_1133', 'B2_1134', 'B2_1135', 'B2_1136', 'B2_1137', 'B2_1138', 'B2_1139', 'B2_1140', 'B2_1141', 'B2_1142', 'B2_1143', 'B2_1144', 'B2_1145', 'B2_1146', 'B2_1147', 'B2_1148', 'B2_1149', 'B2_1150', 'B2_1151', 'B2_1152', 'B2_1153', 'B2_1154', 'B2_1155', 'B2_1156', 'B2_1157', 'B2_1158', 'B2_1159', 'B2_1160', 'B2_1161', 'B2_1162', 'B2_1163', 'B2_1164', 'B2_1165', 'B2_1166', 'B2_1167', 'B2_1168', 'B2_1169', 'B2_1170', 'B2_1171', 'B2_1172', 'B2_1173', 'B2_1174', 'B2_1175', 'B2_1176', 'B2_1177', 'B2_1178', 'B2_1179', 'B2_1180', 'B2_1181', 'B2_1182', 'B2_1183', 'B2_1184', 'B2_1185', 'B2_1186', 'B2_1187', 'B2_1188', 'B2_1189', 'B2_1190', 'B2_1191', 'B2_1192', 'B2_1193', 'B2_1194', 'B2_1195', 'B2_1196', 'B2_1197', 'B2_1198', 'B2_1199', 'B2_1200', 'B2_1201', 'B2_1202', 'B2_1203', 'B2_1204', 'B2_1205', 'B2_1206', 'B2_1207', 'B2_1208', 'B2_1209', 'B2_1210', 'B2_1211', 'B2_1212', 'B2_1213', 'B2_1214', 'B2_1215', 'B2_1216', 'B2_1217', 'B2_1218', 'B2_
```

3. Definir variables de identificación

NFCSID	A1	A2	A3	B2_1	B2_2
--------	----	----	----	------	------

Crea la lista ID_VARS con las columnas que son identificadores únicos 2824.000000 2824.000000 2824.000000 2824.000000 2824.000000 2824.000000

```
ID_VARS = ['NFCSID']
TARGET = 'B10'
WEIGHT_VARS = ['WGT1'] # Puedes quitar si no usas pesos

# Verificar que TARGET existe
if TARGET not in df.columns:
    raise ValueError(f"La columna TARGET '{TARGET}' no se encuentra en el dataset.")
```

75% 2.021027e+09 2.000000 1.000000 1.0 1.000000 2.000000 2.000000

4. Construcción de la matriz de características (X) y vector objetivo (y)

Top 10 variables con más % de NA:
 Elimina de df las columnas de identificación, objetivo y pesos; añade variables derivadas si existen; define y; y muestra las columnas resultantes de X.
NFCSID 0.0

```
# Construir X partiendo de df: quitamos ID, TARGET, WEIGHT_VARS
X = df.drop(columns=ID_VARS + [TARGET] + WEIGHT_VARS, errors='ignore')
```

```
# Forzar que las derivadas estén en X (si existen en df)
derived_vars = ['FL_SCORE', 'PORTFOLIO_DIVERSITY', 'TRADER_SCORE']
for v in derived_vars:
    if v in df.columns and v not in X.columns:
        X[v] = df[v]
```

```
# Definir y
y = df[TARGET]
```

```
# Verificar
print("\n Mostrar columnas del df para verificar:")
print(X.columns.tolist())
print("Total columnas en X:", len(X.columns))
print("Shape de X:", X.shape)
print("Shape de y:", y.shape)
```

proportion

Mostrar columnas del df para verificar:
 ['A1', 'A2', 'A3', 'B2_1', 'B2_2', 'B2_3', 'B2_4', 'B2_5', 'B2_20', 'B2_23', 'B2_24',
 Total columnas en X: 92
 Shape de X: (2824, 92)
 Shape de y: (2824,)

1.0 0.081091

5. División estratificada de los datos

dtype: float64

Realiza una división estratificada en tres pasos: extrae primero el conjunto de prueba (15%), luego el de validación (~15%), y mantiene el restante como entrenamiento; finalmente muestra las formas de cada subconjunto

```
# División estratificada en Train / Validation / Test

# Primer split: 15% test
X_temp, X_test, y_temp, y_test = train_test_split(
    X, y, test_size=0.15, stratify=y, random_state=RANDOM_STATE
)
# Segundo split: ~15% validación
X_train, X_val, y_train, y_val = train_test_split(
    X_temp, y_temp, test_size=0.1765, stratify=y_temp, random_state=RANDOM_STATE
)

print(" Shapes resultantes:")
print(f" • X_train: {X_train.shape}, y_train: {y_train.shape}")
print(f" • X_val: {X_val.shape}, y_val: {y_val.shape}")
print(f" • X_test: {X_test.shape}, y_test: {y_test.shape}")

→ Shapes resultantes:
• X_train: (1976, 92), y_train: (1976,)
• X_val: (424, 92), y_val: (424,)
• X_test: (424, 92), y_test: (424,)
```

6. Instalación de pyarrow y guardado de los splits en formato Parquet

Se asegura de que pyarrow esté instalado y guarda X_train, y_train, X_val, y_val, X_test y y_test en archivos Parquet en la ruta configurada, confirmando al final la operación.

```
# Guardado de splits (VERSIÓN FINAL)

# Asegurarse de que la librería para Parquet esté instalada
!pip install pyarrow -q

# Guardar splits en Drive usando la ruta FINAL desde config.py
X_train.to_parquet(FINAL_SPLITS_DIR / 'X_train.parquet')
y_train.to_frame().to_parquet(FINAL_SPLITS_DIR / 'y_train.parquet')

X_val.to_parquet(FINAL_SPLITS_DIR / 'X_val.parquet')
```