

Laboratorio 8

Rodrigo Mansilla

Brandon Reyes

Carga del Conjunto de Datos:

Dimensiones del dataset: (581012, 55)

	Elevation	Aspect	Slope	Horizontal_Distance_To_Hydrology	Vertical_Distance_To_Hydrology
0	2596.0	51.0	3.0	258.0	0.0
1	2590.0	56.0	2.0	212.0	-6.0
2	2804.0	139.0	9.0	268.0	65.0
3	2785.0	155.0	18.0	242.0	118.0
4	2595.0	45.0	2.0	153.0	-1.0

5 rows × 55 columns

Escalado de variables numéricas

Forma: 581,012 filas × 55 columnas (54 features + Cover_Type).

Numéricas (10): magnitudes muy distintas (elevación ~1859–3858, distancias hasta ~7000, hillshade 0–254) ⇒ bien escalar solo estas.

Binarias (44): Wilderness_Area1..4 y Soil_Type1..40 son 0/1 ⇒ dejarlas sin escalar (passthrough).

Target: Cover_Type = {1..7} con desbalance (clases 2 y 1 dominan; 4 y 5 pequeñas).

Implicación: usar estratificación en los splits y evaluar con métricas por clase (macro-F1).

Exploración y Descripción de Variables

Tipos de datos (muestra):

Elevation float64
Aspect float64
Slope float64
Horizontal_Distance_To_Hydrology float64
Vertical_Distance_To_Hydrology float64
Horizontal_Distance_To_Roadways float64
Hillshade_9am float64
Hillshade_Noon float64
Hillshade_3pm float64
Horizontal_Distance_To_Fire_Points float64
Wilderness_Area1 float64
Wilderness_Area2 float64
Wilderness_Area3 float64
Wilderness_Area4 float64
Soil_Type1 float64

dtype: object

Descripción de las variables numéricas:

	count	mean	std	min	25%	50%
Elevation	581012.0	2959.365301	279.984734	1859.0	2809.0	2996.0
Aspect	581012.0	155.656807	111.913721	0.0	58.0	127.0
Slope	581012.0	14.103704	7.488242	0.0	9.0	13.0
Horizontal_Distance_To_Hydrology	581012.0	269.428217	212.549356	0.0	108.0	218.0
Vertical_Distance_To_Hydrology	581012.0	46.418855	58.295232	-173.0	7.0	30.0
Horizontal_Distance_To_Roadways	581012.0	2350.146611	1559.254870	0.0	1106.0	1997.0
Hillshade_9am	581012.0	212.146049	26.769889	0.0	198.0	218.0
Hillshade_Noon	581012.0	223.318716	19.768697	0.0	213.0	226.0
Hillshade_3pm	581012.0	142.528263	38.274529	0.0	119.0	143.0
Horizontal_Distance_To_Fire_Points	581012.0	1980.291226	1324.195210	0.0	1024.0	1710.0

Valores únicos en columnas binarias (deben ser 0/1):

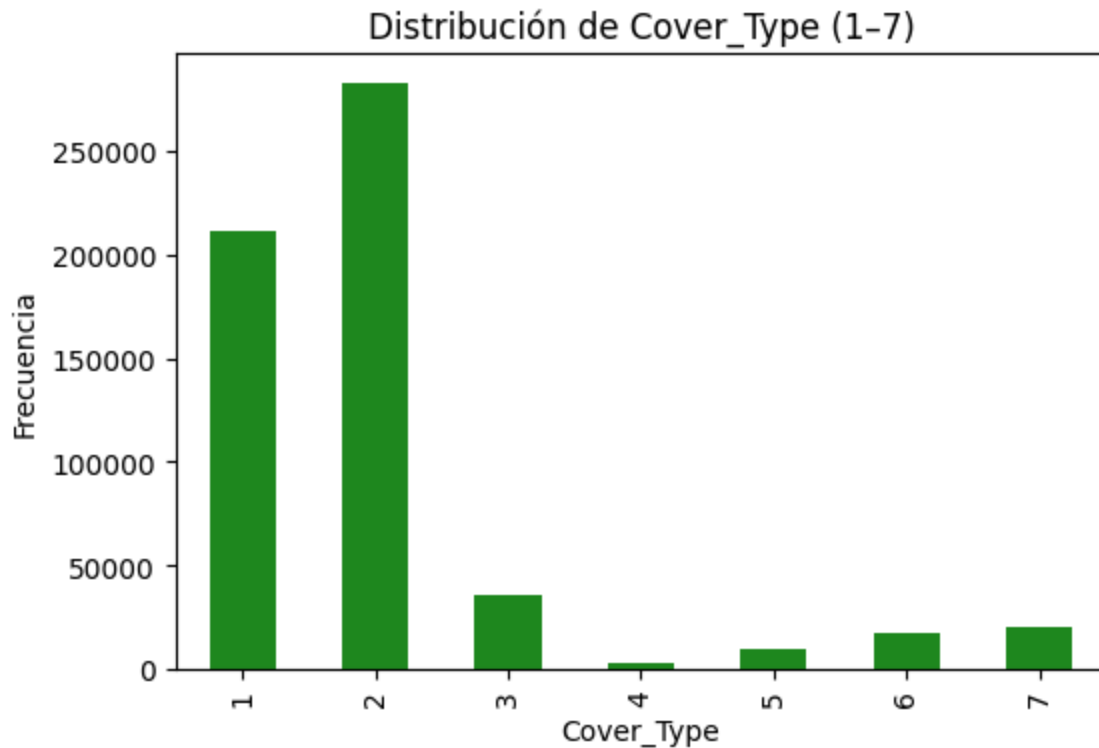
```
{'Wilderness_Area1': [np.float64(0.0), np.float64(1.0)], 'Wilderness_Area2': [np.float64(0.0), np.float64(1.0)]}
{'Soil_Type1': [np.float64(0.0), np.float64(1.0)], 'Soil_Type2': [np.float64(0.0), np.float64(1.0)]}
```

Distribución de la variable objetivo (Cover_Type):

Cover_Type

1 211840
2 283301
3 35754
4 2747
5 9493
6 17367
7 20510

Name: count, dtype: int64



Valores nulos totales: 0

Etiquetar los datos normales y los anormales

Etiqueta Binaria "is_normal" (Normal vs Anómalo)

Conteo de registros por clase:

is_normal

0 297711

1 283301

Name: count, dtype: int64

Proporciones (%):

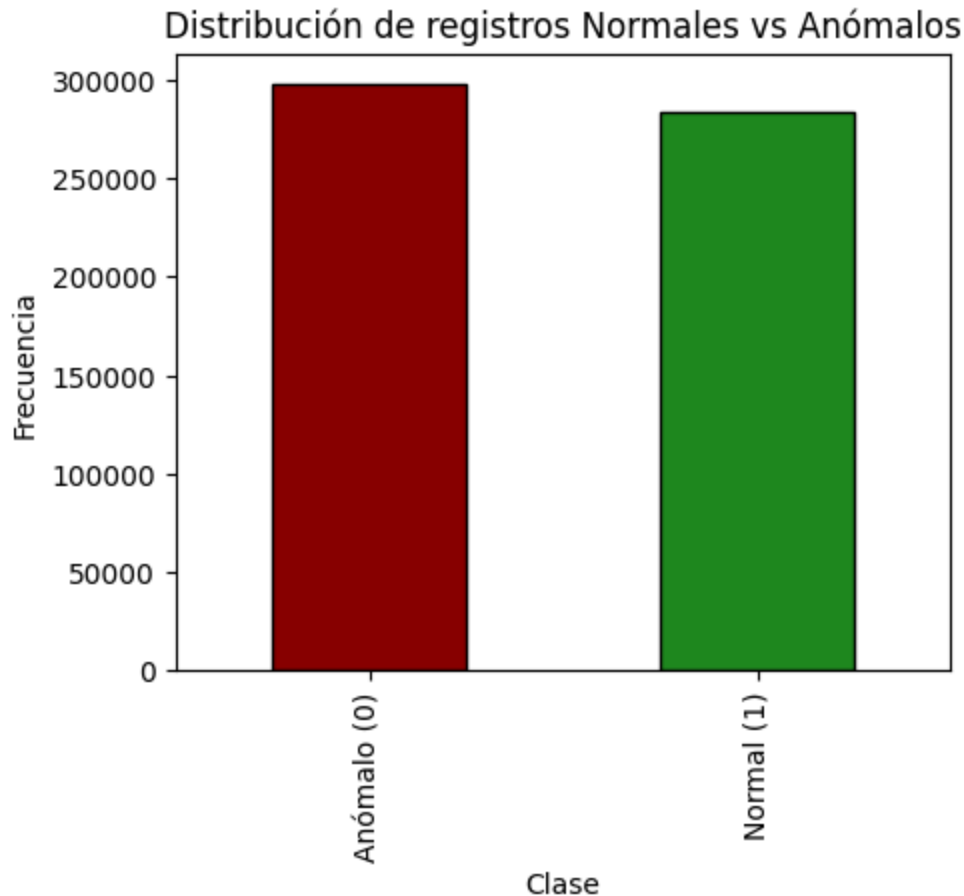
is_normal

0 51.24

1 48.76

Name: proportion, dtype: float64

Visualización del Balance de Clases



División de datos Train/Val/Test:

División General de Datos (Train/Valid/Test)

Tamaño del conjunto de prueba: (116203, 54)

Proporción de normales en TEST: 48.76 %

Separación de Normales para Train y Valid del Autoencoder

Tamaños:

- Train_norm: (181312, 54)
- Valid_norm: (45329, 54)
- Test (mixto): (116203, 54)

¿Por qué entrenar el Autoencoder solo con datos normales?

Un **autoencoder** aprende a reconstruir patrones de entrada minimizando el error de reconstrucción (MSE).

Si se entrena con datos *anómalos*, estos patrones “extraños” también son aprendidos, lo que **dificulta distinguir** entre normalidad y anomalía.

Por tanto:

- **Train/Valid:** deben contener **solo observaciones normales**, para que el modelo capture únicamente la "manifold" normal.
- **Test:** debe incluir tanto normales como anómalos, ya que ahí se evalúa si el error de reconstrucción permite **detectar desviaciones anómalas**.

Modelo de Autocodificador:

Preprocesamiento solo para numéricas escaladas, binarias intactas

Dimensiones transformadas → (181312, 54)

Arquitectura Simétrica

Model: "functional_2"


Layer (type)	Output Shape	Param #
input_layer_2 (InputLayer)	(None , 54)	0
sequential (Sequential)	(None , 16)	18,416
sequential_1 (Sequential)	(None , 54)	18,198


Total params: 36,614 (143.02 KB)


Trainable params: 36,230 (141.52 KB)


Non-trainable params: 384 (1.50 KB)


Entrenamiento con Early Stopping


Epoch 1/60
89/89  19s 56ms/step - loss: 0.1549 - val_loss: 0.1658


Epoch 2/60
89/89  4s 39ms/step - loss: 0.0728 - val_loss: 0.1135


Epoch 3/60
89/89  3s 37ms/step - loss: 0.0550 - val_loss: 0.0712


Epoch 4/60
89/89  4s 40ms/step - loss: 0.0452 - val_loss: 0.0446


Epoch 5/60
89/89  5s 52ms/step - loss: 0.0390 - val_loss: 0.0313


Epoch 6/60
89/89  4s 46ms/step - loss: 0.0349 - val_loss: 0.0256


Epoch 7/60
89/89  5s 43ms/step - loss: 0.0316 - val_loss: 0.0217


Epoch 8/60
89/89  4s 33ms/step - loss: 0.0292 - val_loss: 0.0189


Epoch 9/60
89/89  3s 33ms/step - loss: 0.0271 - val_loss: 0.0167


Epoch 10/60
89/89  3s 34ms/step - loss: 0.0254 - val_loss: 0.0153


Epoch 11/60
89/89  3s 38ms/step - loss: 0.0240 - val_loss: 0.0134


Epoch 12/60
89/89  3s 32ms/step - loss: 0.0229 - val_loss: 0.0128


Epoch 13/60
89/89  3s 38ms/step - loss: 0.0220 - val_loss: 0.0123


Epoch 14/60
89/89  3s 31ms/step - loss: 0.0212 - val_loss: 0.0114


Epoch 15/60
89/89  3s 34ms/step - loss: 0.0204 - val_loss: 0.0123


Epoch 16/60
89/89  3s 31ms/step - loss: 0.0197 - val_loss: 0.0111


Epoch 17/60
89/89  3s 36ms/step - loss: 0.0190 - val_loss: 0.0103


Epoch 18/60
89/89  3s 33ms/step - loss: 0.0184 - val_loss: 0.0105


Epoch 19/60
89/89  3s 38ms/step - loss: 0.0178 - val_loss: 0.0102


Epoch 20/60
89/89  3s 37ms/step - loss: 0.0171 - val_loss: 0.0101


Epoch 21/60
89/89  4s 43ms/step - loss: 0.0166 - val_loss: 0.0096


Epoch 22/60
89/89  5s 60ms/step - loss: 0.0161 - val_loss: 0.0093

Epoch 23/60
89/89  8s 85ms/step - loss: 0.0156 - val_loss: 0.0092

Epoch 24/60
89/89  9s 69ms/step - loss: 0.0150 - val_loss: 0.0085

Epoch 25/60
89/89  7s 75ms/step - loss: 0.0144 - val_loss: 0.0093

Epoch 26/60
89/89  4s 40ms/step - loss: 0.0140 - val_loss: 0.0088

Epoch 27/60
89/89  4s 44ms/step - loss: 0.0135 - val_loss: 0.0088

Epoch 28/60

89/89 ————— 4s 43ms/step - loss: 0.0131 - val_loss: 0.0092

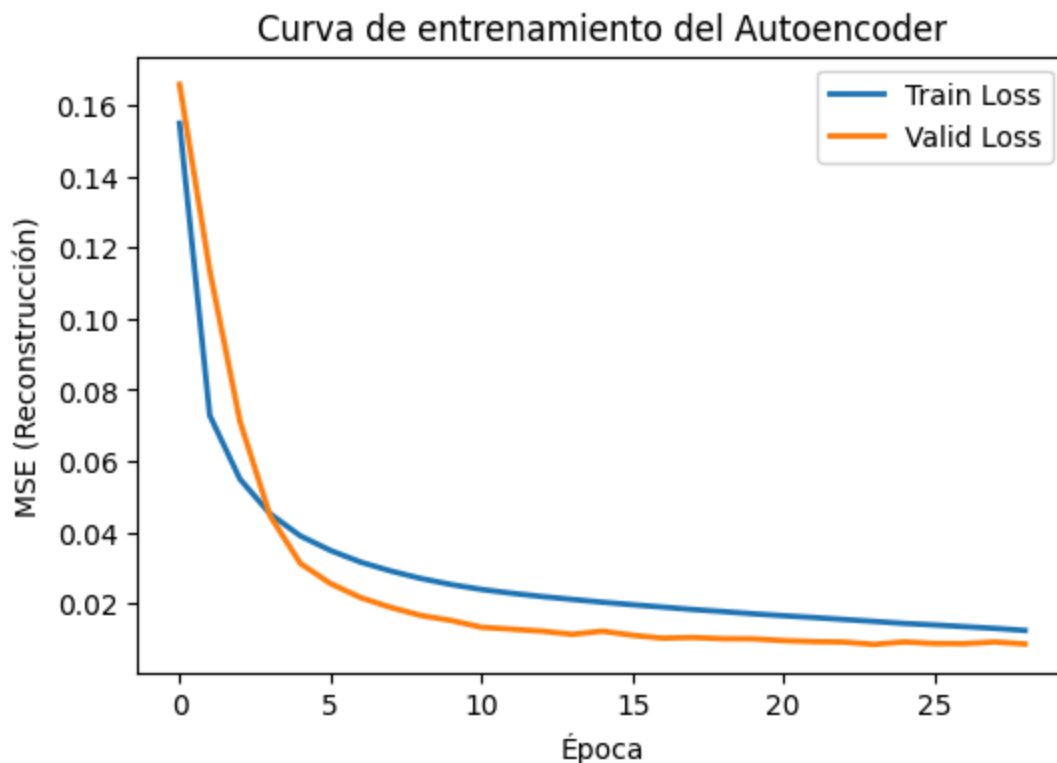
Epoch 29/60

89/89 ————— 3s 37ms/step - loss: 0.0125 - val_loss: 0.0087

Características del modelo

- **Arquitectura simétrica:** encoder y decoder con capas espejo ($128 \rightarrow 64 \rightarrow 32 \rightarrow 16 \rightarrow 32 \rightarrow 64 \rightarrow 128$).
- **Regularización:** Dropout y L2 penalty reducen sobreajuste.
- **Batch Normalization:** estabiliza la convergencia y mejora la generalización.
- **Métrica MSE:** el error de reconstrucción mide cuán "normal" es una observación.
- **Early Stopping:** detiene el entrenamiento si no hay mejora en `val_loss` por 5 épocas
→ evita sobreentrenamiento y limita el tiempo total

Training curve



Modelos de Isolation Forest y LOF:

Consideraciones sobre los modelos de detección no supervisada

- **Isolation Forest** separa anomalías por aislamiento aleatorio de atributos.
 - Escala bien a grandes volúmenes de datos.
 - Ajustar `n_estimators`, `max_samples` y el percentil del umbral permite controlar el balance precisión-recall.
- **Local Outlier Factor** compara la densidad local de un punto con la de sus vecinos.

- Detecta anomalías “de contexto”, pero es sensible a la escala y a `n_neighbors`.
- Ambos se entrenan solo con **normales**, evitando que las anomalías sesguen la distribución aprendida.
- Para comparar con el autoencoder, se utilizan métricas comunes: **F1, ROC-AUC, PR-AUC**.

Isolation Forest

Umbral (percentil 5 `valid_norm`): 0.082752

[IsolationForest] Acc=0.6304 Prec=0.5731 Rec=0.9486 F1=0.7145 | ROC-AUC=0.7250
PR-AUC=0.6650

Matriz de confusión [0:Anómalo, 1:Normal]:

```
[[19512 40031]
 [ 2913 53747]]
```

LOF

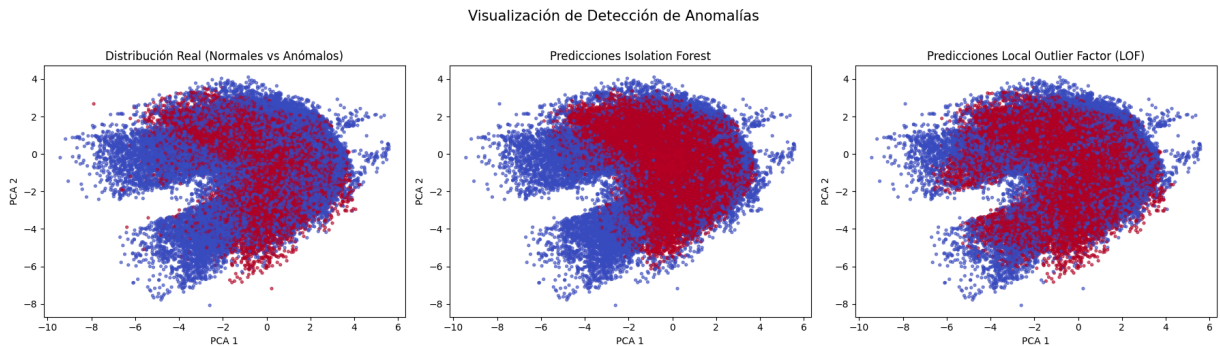
Umbral (p95 `valid_norm`): -0.224086

[LOF] Acc=0.7452 Prec=0.6675 Rec=0.9514 F1=0.7846 | ROC-AUC=0.8598 PR-AUC=0.8259

Matriz de confusión [0:Anómalo, 1:Normal]:

```
[[32691 26852]
 [ 2751 53909]]
```

Visualización 2D de Anomalías



Interpretación

- Cada punto es una observación del conjunto de prueba (reducido a 2D con PCA).
- Colores:
 - Azul → Normal (predicho 1)
 - Rojo → Anómalo (predicho 0)

Panel izquierdo: Etiquetas verdaderas (`y_test`), muestra la distribución real.

Panel central: Predicciones del *Isolation Forest*, que tiende a separar regiones dispersas o extremas.

Panel derecho: Predicciones del *LOF*, más sensible a la densidad local: suele detectar microgrupos aislados como anomalías.

En un buen modelo, los anómalos aparecen como pequeños grupos periféricos o puntos rojos dispersos fuera de las regiones densas azules.

Evaluación de los Modelos:

Matriz de Confusión del Mejor Modelo

```
-----  
NameError                                Traceback (most recent call last)  
Cell In[32], line 4  
      1 model_choice = "AE"    # "AE" | "IF" | "LOF"  
      3 if model_choice == "AE":  
----> 4     y_pred = (err_test <= threshold).astype(int)  
      5     scores = -err_test  
      6     label = "Autoencoder"  
  
NameError: name 'err_test' is not defined
```