

Model Overview (Tercera iteración)

3.1. Variables objetivo y predictores

Utilizamos prácticamente todas las features excepto todas las variables categóricas que son las siguientes:

Python

```
X = df_balanced.drop(['Fecha',
                      'Número del animal',
                      'Estado de la reproducción',
                      'Estado de Producción',
                      'Estado de Producción Codificado'],
                     axis=1).copy()
```

Y nuestra variable objetivo será justo una que acabamos de dropear, que es “Estado de Producción Codificado”

Python

```
y = df_balanced['Estado de Producción Codificado']
```

- **Clase 0:** En producción (previo al secado, 30 días antes)
- **Clase 1:** En monitoreo (en producción durante gestación)
- **Clase 2:** previo a secado (período de descanso)

3.2. Preprocesamiento

Cada modelo implementa un pipeline estandarizado con las siguientes etapas:

1. El pipeline general a groso modo realiza lo siguiente:

Python

```
PREPROCESSING_PIPELINE = {
    'paso_1': SimpleImputer(strategy='median'), # Manejo de
    valores faltantes
    'paso_2': StandardScaler(), # Normalización (excepto RF)
```

```

'paso_3': OneHotEncoder(),
# Solo para
variables categóricas
}

```

1. Codificación de Variables

En este apartado pretendemos cambiar los tipos de datos dependiendo si son variables categóricas o numéricas

Python

```

# Variable objetivo
y_encoded = LabelEncoder().fit_transform(df['Estado de
Producción'])

# Features temporales
df['dia_del_año'] = df['Fecha'].dt.dayofyear
df['sen_dia'] = np.sin(2 * np.pi * df['dia_del_año']/365)
df['cos_dia'] = np.cos(2 * np.pi * df['dia_del_año']/365)

```

1. Normalizaciones por modelo

Modelo	Estrategia de Escalado	Justificación
Regresión Logística	StandardScaler	Requiere normalización para convergencia
Random Forest	No escalado	Árboles son invariantes a escala
XGBoost	No escalado	Optimizado para distribuciones originales
MLP	StandardScaler	Crítico para estabilidad de los gradientes
TabNet	QuantileTransformer	Maneja mejor distribuciones no normales

3.3. Estrategia de entrenamiento

Cada modelo incorpora una función de entrenamiento diferente. Donde se consideran diferentes parámetros e hiperparámetros ya que no todos los modelos son necesarios. Para más detalle se recomienda checar cada notebook con sus respectivos parámetros

3.4. Métricas de evaluación

Métrica	Justificación
Accuracy	Mide el porcentaje total de predicciones totales, útil como referencia general del rendimiento global del modelo.
F1-score	Es la combinación de precisión y recall de manera equilibrada, esto es ideal en caso de que las clases estén desbalanceadas.
recall	Indica qué tan bien el modelo detecta los casos positivos, esto es clave cuando se vuelve más costoso dejar pasar un caso positivo que cometer un falso positivo.
Matriz de confusión	Muestra exactamente las distribuciones de los aciertos y errores del modelo, ayuda para identificar sesgos, tipos de errores y entender realmente el comportamiento.