RETOS: Análisis, Escalamiento y Modelado de Datos Agrícolas

Dataset: Cosechas_Cosechas_2023b.csv

Documento Actualizado con Pasos de Modelado

Fecha: 7 de Agosto de 2025

Total de páginas: 4

1 RETO 1: Análisis, Escalamiento y Modelado de Toneladas Cosechadas

Dataset: Cosechas_Cosechas_2023b.csv
Variable objetivo: Toneladas Cosechadas
Pasos detallados:

1. **Importar las bibliotecas necesarias**:
   Utilizar pandas, numpy, seaborn, matplotlib para procesamiento y visualización. Utilizar sklearn: LabelEncoder, OneHotEncoder, StandardScaler, MinMaxScaler, RobustScaler, PowerTransformer, Normalizer para preprocesamiento. Incluir módulos para modelado: train_test_split, LinearRegression, RandomForestRegressor, GradientBoostingRegressor, mean_squared_error, r2_score, mean_absolute_error.
2. **Cargar el archivo CSV** desde la carpeta `input/`.
3. **Mostrar las primeras filas** con `.head()` para verificar la lectura del dataset.
4. **Revisar estructura del dataset**:
   Usar `.info()`, `.describe()`, `.isnull().sum()` para comprender los tipos de datos y los valores faltantes.
5. **Tratar valores faltantes**:
   Rellenar con la mediana si son numéricos. Rellenar con "Desconocido" o el valor más frecuente si son categóricos.
6. **Convertir tipos de datos**:
   Asegurarse que las fechas estén en formato datetime. Convertir variables categóricas a tipo string si es necesario.
7. **Codificar variables categóricas**:
   Aplicar LabelEncoder a columnas como Especie, Empresa. Aplicar OneHotEncoder a Zona, Periodo Información. Alternativamente, usar `pd.get_dummies()` para codificación.
8. **Aplicar escalamiento a Toneladas Cosechadas**:
   StandardScaler, MinMaxScaler, RobustScaler. Guardar cada resultado como nuevas columnas.
9. **Aplicar transformaciones matemáticas a Toneladas Cosechadas**:
   Logarítmica (`np.log1p`). Raíz cuadrada (`np.sqrt`). Box-Cox con `PowerTransformer(method='ye`
10. **Aplicar normalización (L2)** a Toneladas Cosechadas.
11. **Dividir los datos**:
    Separar el dataset en conjuntos de entrenamiento (80 %) y prueba (20 %) utilizando `train_test_split`.
12. **Aplicar modelos de regresión**:
    Entrenar LinearRegression, RandomForestRegressor y GradientBoostingRegressor sobre los datos de entrenamiento.
13. **Comparar modelos y seleccionar el mejor**:
    Evaluar los modelos utilizando Mean Squared Error (MSE), Rš y Mean Absolute

Error (MAE) en el conjunto de prueba. Seleccionar el modelo con el menor MSE, considerando Rš y MAE como métricas secundarias.

14. **Visualizar distribuciones escaladas** con `seaborn.kdeplot` para las variables transformadas.

15. **Crear una tabla resumen** con:
    Media, desviación estándar, por cada técnica de escalamiento y transformación aplicada.

16. **Generar interpretación automática en HTML** con comentarios para cada técnica y resultados de los modelos.

17. **Construir un dashboard HTML** que incluya:
    Gráfico de distribuciones, tabla resumen, interpretaciones dinámicas de transformaciones y modelado.

18. **Guardar resultados**:
    Dataset procesado en `output/`. Tabla resumen en `.csv`. Dashboard en `.html`. Imagen del gráfico en `.png`.

2 RETO 2: Análisis, Escalamiento y Modelado de Rendimiento (kg/mš)

Dataset: Cosechas_Cosechas_2023b.csv
Variable objetivo: Rendimiento (kg/mš)
Pasos detallados:

1. **Importar bibliotecas** requeridas para procesamiento, visualización y modelado.
2. **Cargar el archivo** desde `input/` y visualizar las primeras filas con `.head()`.
3. **Analizar el dataset** con `.info()`, `.describe()` y revisar valores nulos.
4. **Limpieza de datos faltantes** en la variable Rendimiento (kg/mš) y otras columnas relevantes.
5. **Conversión de tipos de datos** (fechas a datetime, strings, numéricos).
6. **Codificación de variables categóricas**:
   LabelEncoder: Especie, Centro. OneHotEncoder: Zona, Periodo Información.
7. **Escalamiento de Rendimiento (kg/mš)** usando:
   StandardScaler, MinMaxScaler, RobustScaler.
8. **Transformaciones matemáticas**:
   Logarítmica (`np.log1p`). Raíz cuadrada (`np.sqrt`). Box-Cox con `PowerTransformer(method='ye`
9. **Normalización (L2)** de la variable.
10. **Dividir los datos**:
    Separar el dataset en conjuntos de entrenamiento (80 %) y prueba (20 %) utilizando `train_test_split`.
11. **Aplicar modelos de regresión**:
    Entrenar LinearRegression, RandomForestRegressor y GradientBoostingRegressor sobre los datos de entrenamiento.
12. **Comparar modelos y seleccionar el mejor**:
    Evaluar los modelos utilizando Mean Squared Error (MSE), Rš y Mean Absolute Error (MAE) en el conjunto de prueba. Seleccionar el modelo con el menor MSE, considerando Rš y MAE como métricas secundarias.
13. **Visualización KDE** de cada técnica aplicada a Rendimiento.
14. **Tabla resumen** de media y desviación estándar para cada transformación.
15. **Interpretación automática HTML** para transformaciones y resultados de modelado.
16. **Dashboard HTML** con gráficos, tabla resumen y interpretaciones.
17. **Guardar resultados**:
    Dataset transformado en `output/`. Tabla resumen en `.csv`. Gráfico en `.png`. Dashboard en `.html`.

3 RETO 3: Análisis, Escalamiento y Modelado de Superficie Cosechada

Dataset: Cosechas_Cosechas_2023b.csv
Variable objetivo: Superficie Cosechada
Pasos detallados:

1. **Importar bibliotecas** necesarias para procesamiento, visualización y modelado.
2. **Leer el dataset** desde la carpeta `input/` y mostrar las primeras filas con `.head()`.
3. **Explorar el dataset** con `.info()`, `.describe()` y revisar valores nulos.
4. **Imputar valores faltantes** en Superficie Cosechada y otras columnas relevantes.
5. **Corregir tipos de datos** si es necesario (fechas a datetime, strings, numéricos).
6. **Codificación de variables categóricas**:
   LabelEncoder: Titular/Operador, Especie. OneHotEncoder: Región, Zona.
7. **Aplicar técnicas de escalamiento** sobre Superficie Cosechada:
   StandardScaler, MinMaxScaler, RobustScaler.
8. **Transformaciones matemáticas**:
   Logarítmica (`np.log1p`). Raíz cuadrada (`np.sqrt`). Box-Cox con `PowerTransformer(method='ye`
9. **Aplicar normalización (L2)** a Superficie Cosechada.
10. **Dividir los datos**:
    Separar el dataset en conjuntos de entrenamiento (80 %) y prueba (20 %) utilizando `train_test_split`.
11. **Aplicar modelos de regresión**:
    Entrenar LinearRegression, RandomForestRegressor y GradientBoostingRegressor sobre los datos de entrenamiento.
12. **Comparar modelos y seleccionar el mejor**:
    Evaluar los modelos utilizando Mean Squared Error (MSE), Rš y Mean Absolute Error (MAE) en el conjunto de prueba. Seleccionar el modelo con el menor MSE, considerando Rš y MAE como métricas secundarias.
13. **Graficar las distribuciones** de las variables transformadas con `seaborn.kdeplot`.
14. **Generar tabla resumen** con media y desviación estándar por técnica.
15. **Crear interpretación automática en HTML** para transformaciones y resultados de modelado.
16. **Diseñar dashboard completo** con gráficos, tabla resumen y interpretaciones.
17. **Guardar resultados**:
    Dataset procesado en `output/`. Tabla resumen en `.csv`. Dashboard en `.html`. Imagen del gráfico en `.png`.

4 RETO 4: Análisis, Escalamiento y Modelado de Mortalidad Acumulada (kg)

Dataset: Cosechas_Cosechas_2023b.csv
Variable objetivo: Mortalidad Acumulada (kg)
Pasos detallados:

1. **Importar módulos** de procesamiento, visualización y modelado.
2. **Leer archivo CSV** desde `input/` y mostrar datos con `.head()`.
3. **Estudiar la variable Mortalidad Acumulada (kg)** y su distribución inicial con `.describe()`.
4. **Tratar valores nulos o atípicos** en Mortalidad Acumulada (kg) y otras columnas relevantes.
5. **Convertir formatos de columnas** si es necesario (fechas a datetime, strings, numéricos).
6. **Codificar variables categóricas** relacionadas como Centro, Especie, Región: Usar LabelEncoder o OneHotEncoder según corresponda.
7. **Aplicar escalamiento a Mortalidad Acumulada (kg)**: StandardScaler, MinMaxScaler, RobustScaler.
8. **Realizar transformaciones matemáticas**: Logarítmica (`np.log1p`). Raíz cuadrada (`np.sqrt`). Box-Cox con `PowerTransformer(method='ye`
9. **Aplicar normalización L2** a la variable.
10. **Dividir los datos**: Separar el dataset en conjuntos de entrenamiento (80 %) y prueba (20 %) utilizando `train_test_split`.
11. **Aplicar modelos de regresión**: Entrenar LinearRegression, RandomForestRegressor y GradientBoostingRegressor sobre los datos de entrenamiento.
12. **Comparar modelos y seleccionar el mejor**: Evaluar los modelos utilizando Mean Squared Error (MSE), Rš y Mean Absolute Error (MAE) en el conjunto de prueba. Seleccionar el modelo con el menor MSE, considerando Rš y MAE como métricas secundarias.
13. **Visualizar las escalas** con `seaborn.kdeplot` para las variables transformadas.
14. **Calcular media y desviación estándar** por técnica en una tabla resumen.
15. **Generar HTML dinámico** con interpretaciones de transformaciones y resultados de modelado.
16. **Diseñar dashboard** con Bootstrap, incluyendo gráfico, tabla resumen y interpretaciones.

17. **Guardar resultados**:
    Dataset final en `output/`. Tabla resumen en `.csv`. Imagen del gráfico en `.png`. Dashboard en `.html`.