Proyecto de probabilidad

Luis Sandoval 26781082, Gabriel Rodriguez 30172571, Jose Rivero 28492353

2025-04-11

Descripcion del proyecto

El análisis se realizó utilizando el lenguaje de programación **Python (versión 3.12)** y librerías estándar de ciencia de datos como pandas, numpy y scikit-learn. Los datos de entrada provienen del archivo data/Ab19selec.csv, que contiene registros históricos del proceso. Se implementaron y compararon diversas técnicas de regresión, incluyendo Regresión Lineal MCO, Ridge, Lasso, Elastic Net y Selección Secuencial de Características (Sequential Feature Selection). Un ejemplo de ejecucion es el siguiente:

```
python main.py
```

Se deben instalar tambien las dependencias del archivo requirements.txt.

Se utilizo la semilla 2022 en numpy y en random para la generacion de datos reproducibles.

```
seed = 2022
np.random.seed(seed)
random.seed(seed)
```

Comprensión del Problema

Objetivo: Modelar el consumo específico de ClO2 (kg/ADT) en el proceso de blanqueo de pulpa para reducir costos operativos

Comprensión de los Datos

- 98 variables numéricas predictoras
- 1 variable numerica objetivo: kg.ADT (Nombre tecnico)

Métricas:

- **MSE (Error Cuadrático Medio):** Mide el promedio de los errores al cuadrado entre los valores predichos y los reales. Penaliza más los errores grandes. Un valor más bajo indica mejor ajuste. Sus unidades son el cuadrado de las unidades de la variable objetivo ((kg/ADt)^2).
- **R^2 (Coeficiente de Determinación):** Indica la proporción de la varianza en la variable objetivo que es predecible a partir de las variables independientes. Varía entre 0 y 1 (o puede ser negativo para modelos muy malos). Un valor cercano a 1 indica que el modelo explica una gran parte de la variabilidad de los datos.

Limpieza de Datos:

Eliminación de variables no relevantes para el objetivo:

- Variables mecánicas (torques, producción)
- Mediciones redundantes (niveles duplicados, PH indirectos)

- columnas vacias

Preparación para Modelado:

División 80%-20% (entrenamiento-test)

Se estandarizo (o normalizo) la data con StandardScaler para evitar preferencias en variables

con valores muy altos por estar en escalas diferentes.

Modelos Implementados

Regresión Lineal (MCO)

Objetivo: Asume una relación lineal entre las variables predictoras y la variable objetivo. Estima los coeficientes minimizando la suma de los errores al cuadrado. Es simple e interpretable, pero puede ser sensible a la multicolinealidad (alta correlación entre predictores) y no realiza selección de variables.

```
LinearRegression().fit(X_train, y_train)
```

Configuración:

Regresión Ridge

Objetivo: Es una técnica de regularización que añade una penalización L2 (proporcional a la suma de los cuadrados de los coeficientes) a la función de costo de MCO. Esto "encoge" los coeficientes, especialmente los de variables correlacionadas, haciéndolos más pequeños y estables, pero sin llevarlos exactamente a cero. El parámetro alpha controla la intensidad de esta penalización. A veces alfa es llamado tambien lambda.

Configuración: alpha = 1.0 (óptimo por validación cruzada)

```
RidgeCV(alphas=np.logspace(-6,6,13))
```

RidgeCV a diferencia del metodo Ridge normal de scikit busca el valor optimo de alpha a traves de validacion cruzada en el rango de valores especificado en parametros.

Lasso Regression

Objetivo: Aplica una penalización L1 (proporcional a la suma de los valores absolutos de los coeficientes). Esta penalización tiene el efecto de reducir algunos coeficientes exactamente a cero, eliminando efectivamente esas variables del modelo. Es útil para simplificar modelos y mejorar la interpretabilidad cuando se sospecha que muchas variables no son relevantes. El parámetro alpha controla la intensidad de la penalización.

Configuración: alpha = 0.0006 (óptimo por CV)

```
LassoCV(n_alphas=100, max_iter=20000)
```

Al igual que RidgeCV, LassoCV busca el valor optimo de alfa por validacion cruzada. Se permiten un maximo de 20000 iteraciones para que el modelo converja

Elastic Net

Objetivo: Es un modelo híbrido que aplica una combinación de penalizaciones L1 y L2. Está controlado por alpha (intensidad total de regularización) y l1_ratio (proporción de la penalización L1; l1_ratio=1 es Lasso, l1_ratio=0 es Ridge). Es útil en escenarios con alta correlación entre predictores donde Lasso podría seleccionar arbitrariamente una variable de un grupo correlacionado.

Configuración: alpha = 0.0006, l1_ratio = 1.0

Ratios probados: [0.1, 0.5, 0.7, 0.9, 0.95, 0.99, 1]

```
ElasticNetCV(l1_ratios=[...], n_alphas=100)
```

Se busca tambien el valor optimo de alfa y se itera por valores l1 entre una lista de ratios.

Stepwise Regression

Objetivo: Este no es un modelo de regresión como tal, sino una técnica de selección de características. Se utilizó SequentialFeatureSelector en modo "adelante" (direction='forward'). Comenzando sin predictores, el algoritmo añade iterativamente la variable que proporciona la mayor mejora en la métrica.

Configuración: Seleccion automatica de caracteristicas. Se utilizo CV = 5 para el cross validation

```
selector = SequentialFeatureSelector(
      base_model,
      direction="forward",  # Forward
      scoring="r2",
      cv=5,
      n_features_to_select="auto"  # Selección automática
  )

  selector.fit(X_train, y_train)

  selected_features = selector.get_support(indices=True)

  modelo = LinearRegression()

  modelo.fit(X_train[:, selected_features], y_train)
```

Resultados obtenidos

Modelo	MSE	R^2
Regresión Lineal	0.23	0.81
Ridge	0.23	0.81
Lasso	0.23	0.80
Elastic Net	0.23	0.80
Stepwise	0.24	0.80

Selección del Modelo

Aunque todos los modelos muestran un rendimiento similar (R^2 +/- 0.80-0.81), se recomienda emplear ridge debido a:

- Mayor estabilidad numérica.
- Mismo rendimiento que MCO (R^2 = 0.81)
- Resistencia probada a multicolinealidad
- Mantiene todas las variables consideradas relevantes tras la limpieza inicial, lo cual puede ser valioso si se asume que todas ellas aportan alguna información útil al sistema