

Descripción por bloques de código de PredicKNNRegCMetricasGenSolarEolicaCargaEspanaOkGit.R

Este código realiza **pronósticos de variables energéticas (generación solar, eólica y demanda total)** utilizando el algoritmo **K-Nearest Neighbors (K-NN)** en R. A continuación, se detalla su funcionamiento:

1. Preparación de Datos

- **Paquetes y Librerías:**
 - Instala y carga paquetes para manipulación de datos (dplyr, tidyr), gráficos (ggplot2), manejo de fechas (lubridate), y KNN (FNN para regresión K-NN).
- **Lectura de Datos:** Importa el dataset desde GitHub, similar al código anterior, y parsea la columna time a formato datetime usando ymd_hms().

2. Configuración del Modelo

- **Variables Objetivo:**
 - generation.solar, generation.wind.onshore, total.load.actual.
- **Parámetros Clave:**
 - window_size = 168: Ventana histórica de 7 días (168 horas).
 - k_value = 5: Número de vecinos cercanos a considerar en KNN.
 - test_size = 30*24: Últimos 30 días para evaluación.
 - forecast_horizon = 30*24: Pronóstico futuro de 30 días.

3. Funciones Clave

- create_features():
 - Transforma la serie temporal en ventanas deslizantes (igual que en SVM). Por ejemplo, con window_size = 3, convierte [x1, x2, x3, x4] en:

Features: [x1, x2, x3] → Target: x4

- calculate_metrics():

- Calcula métricas: **R²**, **RMSE**, y **MAE** (igual que en SVM).

4. Flujo de Trabajo Principal

a. Evaluación del Modelo (Conjunto de Prueba)

1. **División de Datos:** Separa la serie en entrenamiento (todos menos los últimos 30 días) y prueba (últimos 30 días).
2. **Pronóstico Recursivo con KNN:**
 - Para cada paso en el conjunto de prueba:
 - Usa la función `knn.reg()` del paquete `FNN` para encontrar los **5 vecinos más cercanos** (parámetro `k`) en los datos de entrenamiento.
 - Predice el siguiente valor como el **promedio ponderado de los vecinos**.
 - Actualiza la ventana histórica con la predicción para el siguiente paso.
3. **Cálculo de Métricas:** Compara las predicciones con los valores reales del conjunto de prueba.

b. Pronóstico Futuro

1. **Entrenamiento con Todos los Datos:** Utiliza toda la serie para construir la matriz de características (`full_data`).
2. **Pronóstico de 30 Días:**
 - Aplica el mismo método recursivo, pero usando toda la serie como referencia para buscar vecinos cercanos.
 - Genera predicciones hora a hora para 30 días futuros.

5. Visualización

- Crea gráficos temporales para cada variable pronosticada usando `ggplot2`, mostrando las predicciones futuras.

Diferencias Clave vs. SVM

Aspecto	SVM	KNN
Tipo de Modelo	Modelo paramétrico (ajusta una función no lineal).	Modelo no paramétrico (basado en instancias).
Entrenamiento	Entrena una función explícita (kernel radial).	No hay fase de entrenamiento: las predicciones se calculan "on-the-fly".
Costo Computacional	Costoso durante el entrenamiento, pero rápido en predicción.	Costoso durante la predicción (busca vecinos en todo el conjunto de entrenamiento).
Hiperparámetros	Kernel, parámetros de regularización (C, gamma).	Número de vecinos (k).

Ejemplo de Salida

- **Métricas de Evaluación** (ejemplo):

Variable: generation.solar

R2 RMSE MAE

0.82 130.5 100.2

- **Gráficos:** Líneas temporales con los pronósticos futuros para cada variable.

Contexto y Limitaciones

- **Aplicaciones:**
 - Útil para pronósticos a corto plazo en sistemas energéticos con patrones estacionales.
 - Menos sensible a valores atípicos que SVM (depende de los vecinos cercanos).
- **Limitaciones:**

- **Computacionalmente intensivo** para series largas (la búsqueda de vecinos es $O(n)$).
- Requiere **normalización de datos** si las escalas de las características son diferentes (no se aplica aquí, pues todas las características son la misma variable en ventanas de tiempo).
- La elección de k es crítica: un k pequeño puede sobreajustar, un k grande puede suavizar demasiado.

Mejoras Potenciales

1. **Optimizar Hiperparámetros:** Usar validación cruzada para elegir el mejor k .
2. **Reducir Costo Computacional:** Implementar técnicas como **KD-Trees** o **Ball Trees** (disponibles en otros paquetes como `knnn`).
3. **Incorporar Variables Externas:** Añadir datos meteorológicos (ej. radiación solar para generación solar).

Este código adapta la metodología de pronóstico recursivo al algoritmo KNN, ofreciendo una alternativa simple pero poderosa para modelos de series temporales en energía.