

Descripción por bloques de código de PredicSVMRegCMetricasGenSolarEolicaCargaEspanaOkGit.R

Este código realiza **pronósticos de series temporales para variables energéticas (generación solar, eólica y demanda total)** utilizando **Support Vector Machines (SVM)** en R. Aquí está el desglose detallado:

1. Preparación de Datos

- **Paquetes y Librerías:** Instala y carga paquetes esenciales para manipulación de datos (dplyr, tidyr), gráficos (ggplot2), manejo de fechas (lubridate), y el modelo SVM (e1071).
- **Lectura de Datos:** Importa un dataset de energía desde un repositorio de GitHub. El dataset contiene una columna time con fechas y horas, y múltiples variables relacionadas con generación y consumo de energía.
- **Preprocesamiento de Fechas:** Convierte la columna time a formato datetime usando ymd_hms() y ordena los datos cronológicamente.

2. Configuración del Modelo

- **Variables Objetivo:**
 - generation.solar (generación solar).
 - generation.wind.onshore (generación eólica terrestre).
 - total.load.actual (demanda total de energía).
- **Parámetros Clave:**
 - window_size = 168: Usa las últimas 168 horas (7 días) como ventana de datos históricos para predecir el siguiente valor.
 - test_size = 30*24: Evalúa el modelo en los últimos 30 días de datos.
 - forecast_horizon = 30*24: Realiza pronósticos futuros para 30 días.

3. Funciones Clave

- create_features():

- Transforma la serie temporal en un conjunto de características (**features**) usando ventanas deslizantes. Por ejemplo, si `window_size = 3`, convierte `[x1, x2, x3, x4]` en:

Features: `[x1, x2, x3]` → Target: `x4`

- `calculate_metrics()`:
 - Calcula métricas de evaluación: **R^2 (coeficiente de determinación)**, **RMSE (error cuadrático medio)** y **MAE (error absoluto medio)**.

4. Flujo de Trabajo Principal

a. Evaluación del Modelo (Conjunto de Prueba)

1. **División de Datos:** Separa la serie en entrenamiento (todos excepto los últimos 30 días) y prueba (últimos 30 días).
2. **Entrenamiento del SVM:**
 - Usa un kernel radial (`kernel = "radial"`) para capturar relaciones no lineales.
 - Entrena con ventanas históricas para predecir el siguiente valor.
3. **Pronóstico Recursivo:**
 - Predice paso a paso en el conjunto de prueba, actualizando la ventana con cada predicción para simular un escenario realista.
4. **Cálculo de Métricas:** Evalúa el rendimiento comparando predicciones con valores reales del conjunto de prueba.

b. Pronóstico Futuro

1. **Entrenamiento con Todos los Datos:** Reentrena el modelo usando toda la serie disponible.
2. **Pronóstico de 30 Días:**
 - Genera predicciones recursivas para las próximas 720 horas (30 días) partiendo de la última ventana conocida.
 - Almacena los resultados en `forecast_df`.

5. Visualización

- Crea gráficos separados para cada variable pronosticada usando ggplot2, mostrando las predicciones futuras contra el tiempo.

Ejemplo de Salida

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

Variable: generation.solar

R2	RMSE	MAE
----	------	-----

0.85	120.3	95.6
------	-------	------

- **Gráficos:** Líneas temporales que muestran la tendencia histórica y el pronóstico para cada variable.

Contexto de Aplicación

Este código es útil para:

- Predecir la generación de energías renovables (solar/eólica) y la demanda eléctrica.
- Gestionar operaciones en sistemas de energía con anticipación.
- Analizar patrones estacionales o horarios en los datos.

Limitaciones:

- La precisión depende de la calidad y estacionalidad de los datos.
- SVM puede ser computacionalmente costoso para series muy largas.