Curso básico de Python y Machine Learning con datos atmosféricos

Juan Manuel Sánchez Cerritos

¿Qué es Machine Learning?

Machine Learning (ML) es un campo de la inteligencia artificial que se enfoca en desarrollar sistemas capaces de aprender y mejorar automáticamente a partir de la experiencia, sin ser programados explícitamente para cada tarea.

¿Qué es Machine Learning?

Machine Learning (ML) es un campo de la inteligencia artificial que se enfoca en desarrollar sistemas capaces de aprender y mejorar automáticamente a partir de la experiencia, sin ser programados explícitamente para cada tarea.

- Utiliza algoritmos y modelos matemáticos para identificar patrones en los datos.
- Realiza predicciones o decisiones basadas en estos patrones.
- Se diferencia de la programación tradicional: aprende las reglas directamente de los datos.

Tipos de Machine Learning

1. **Aprendizaje Supervisado**
   - Entrenamiento con datos etiquetados.
   - Ejemplos: Clasificación (spam o no), Regresión (precio de vivienda).

Tipos de Machine Learning

1. **Aprendizaje Supervisado**
   - Entrenamiento con datos etiquetados.
   - Ejemplos: Clasificación (spam o no), Regresión (precio de vivienda).
2. **Aprendizaje No Supervisado**
   - Identificación de patrones en datos no etiquetados.
   - Ejemplos: Clustering, Reducción de dimensionalidad.

Tipos de Machine Learning

1. **Aprendizaje Supervisado**
   - Entrenamiento con datos etiquetados.
   - Ejemplos: Clasificación (spam o no), Regresión (precio de vivienda).
2. **Aprendizaje No Supervisado**
   - Identificación de patrones en datos no etiquetados.
   - Ejemplos: Clustering, Reducción de dimensionalidad.
3. **Aprendizaje por Refuerzo**
   - Agente aprende a través de recompensas o penalizaciones.
   - Ejemplo: Enseñar a un robot a caminar.

Componentes de un Modelo de ML

- **Datos**: Calidad y cantidad determinan el desempeño del modelo.
- **Algoritmo**: Reglas para aprender de los datos (e.g., redes neuronales).
- **Función de Costo**: Mide el error, objetivo es minimizarlo.
- **Entrenamiento y Validación**:
  - **Entrenamiento**: Ajuste de parámetros para aprender.
  - **Validación**: Evaluar el desempeño con nuevos datos.

Clasificación y Regresión

Clasificación:

- Método supervisado para predecir categorías.
- Ejemplo: Clasificar correos como spam o no.

Clasificación y Regresión

Clasificación:

- Método supervisado para predecir categorías.
- Ejemplo: Clasificar correos como spam o no.

Regresión:

- Analiza la relación entre variables dependientes e independientes.
- Ejemplo: Predecir el precio de una vivienda.

Métricas de Evaluación en Regresión

Métricas de Evaluación en Regresión

1. **Coeficiente de Determinación (R^2)**:

$$R^2 = 1 - \frac{\sum(y - \hat{y})^2}{\sum(y - \bar{y})^2}$$

Métricas de Evaluación en Regresión

1. **Coeficiente de Determinación (R^2)**:

$$R^2 = 1 - \frac{\sum(y - \hat{y})^2}{\sum(y - \bar{y})^2}$$

2. **Error Cuadrático Medio (RMSE)**:

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}$$

Métricas de Evaluación en Regresión

1. **Coeficiente de Determinación (R^2)**:

$$R^2 = 1 - \frac{\sum(y - \hat{y})^2}{\sum(y - \bar{y})^2}$$

2. **Error Cuadrático Medio (RMSE)**:

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}$$

3. **Error Absoluto Medio (MAE)**:

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i|$$

Regresión Lineal

Regresión Lineal

Modelo:

$$y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_n X_n + \epsilon$$

- β_0: Intercepto.
- $\beta_1, \beta_2, \ldots, \beta_n$: Coeficientes de regresión.
- ϵ: Error residual.

Biblioteca Scikit-learn

Biblioteca Scikit-learn

Scikit-learn:

- Biblioteca de Python para aprendizaje automático.
- Algoritmos incluidos: Random Forest, Gradient Boosting, K-means, etc.
- Diseñada para interoperar con NumPy y SciPy.

Random Forest

Random Forest

Definición: Modelo de conjunto basado en múltiples árboles de decisión.

Random Forest

Definición: Modelo de conjunto basado en múltiples árboles de decisión.

- Usa **Bootstrap Sampling** y subconjuntos aleatorios de características.
- En regresión: Promedia las predicciones de todos los árboles.
- Reduce el sobreajuste y es robusto a datos ruidosos.

Random Forest

En un modelo de Random Forest, los **hiperparámetros** son configuraciones que controlan cómo se entrena y se construyen los árboles dentro del bosque. Ajustar estos hiperparámetros puede mejorar significativamente el rendimiento del modelo.

- **max_depth**: Profundidad máxima de los árboles.
- **n_estimators**: Número de árboles en el bosque.
- **random_state**: Semilla para generar números aleatorios, utilizada para garantizar la reproducibilidad de los resultados.

Support Vector Regression (SVR)

SVR

- Forma rápida de interpolar datos
- Los datos de entrada son vistos como un vector p-dimensional (una lista ordenada de p números). La SVM busca un hiperplano que separe de forma óptima a los puntos de una clase con respecto a otra

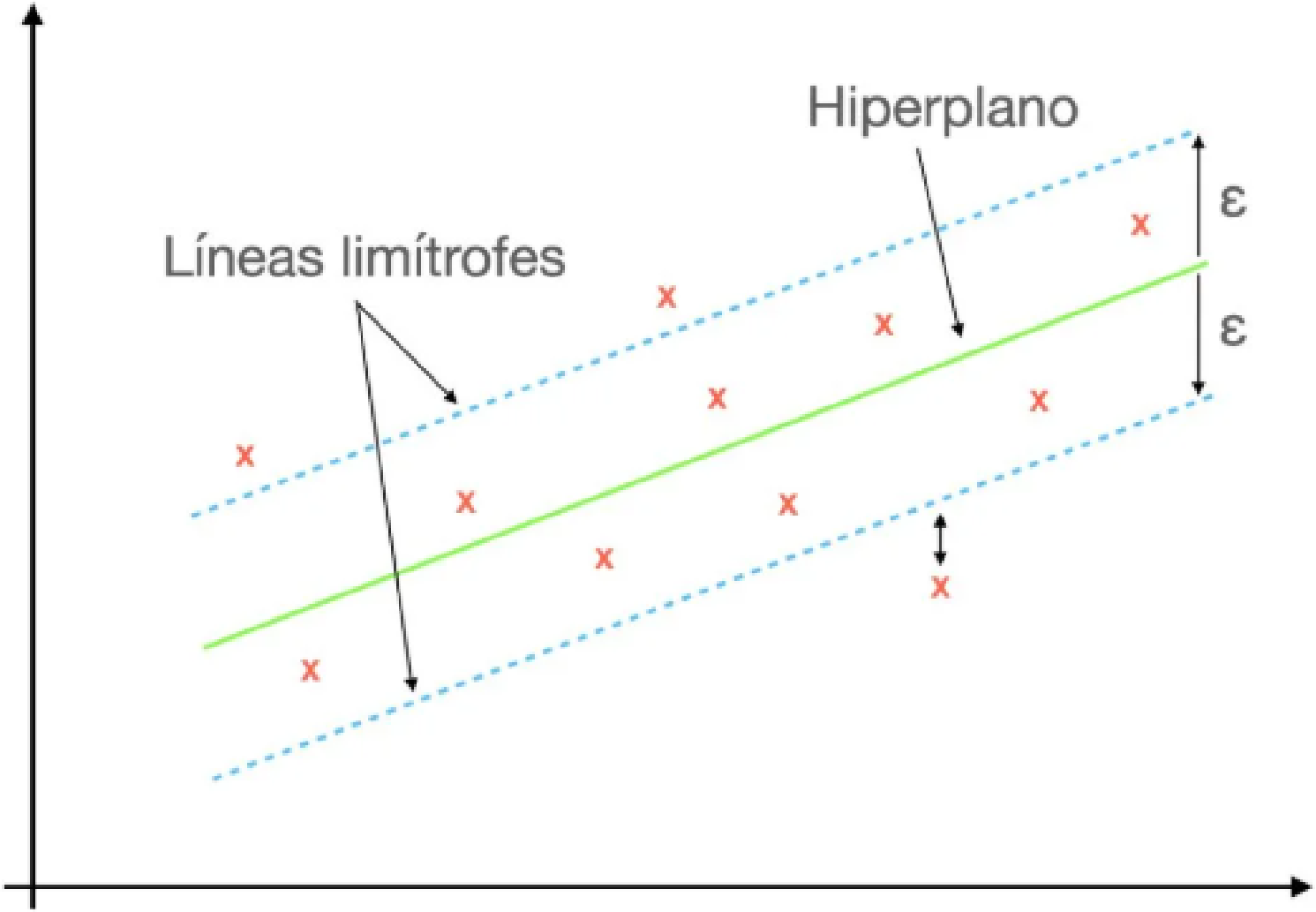

Figure: SVR.

SVR

Estandarización de los datos

Sensibilidad a la escala:

Los kernels, como RBF, dependen de las distancias entre puntos en el espacio de características. Si las características tienen escalas muy diferentes, esto puede causar que el modelo dé mayor peso a ciertas características.

Parámetros del modelo SVR

SVR (Support Vector Regression) tiene varios parámetros importantes que controlan su funcionamiento:

- **kernel='rbf'**:
  - Es la *función de núcleo* que transforma los datos.
  - **RBF (Radial Basis Function)**:
    - Captura relaciones no lineales.
    - Calcula la similitud entre puntos usando una función gaussiana:

$$K(x_i, x_j) = \exp(-\gamma ||x_i - x_j||^2)$$

    - Ideal para problemas con datos complejos.
- **C=1.0**:
  - Controla el equilibrio entre ajuste y regularización.
  - Valores bajos (< 1.0): Más tolerancia a errores, mejor generalización.
  - Valores altos (> 1.0): Penalización estricta de errores, puede sobreajustar.
- **epsilon=0.1**:
  - Define un margen donde los errores no se penalizan.
  - Valores bajos: Más sensibilidad al error, riesgo de sobreajuste.
  - Valores altos: Ignora errores pequeños, genera modelos más simples.

Resumen del impacto de los parámetros

Resumen práctico:

- **kernel='rbf' (linear, poly, sigmoid)**: Captura relaciones no lineales entre las características.
- **C=1.0**: Balance entre ajuste al entrenamiento y generalización.
- **epsilon=0.1**: Controla la tolerancia al error en las predicciones.

Ajustes recomendados:

- Si el modelo subajusta:
  - Incrementar C.
  - Reducir ϵ.
- Si el modelo sobreajusta:
  - Reducir C.
  - Incrementar ϵ.

Aumento de variables temporales

Aumento de Variables Temporales

Aumento de variables temporales consiste en extraer información relevante a partir de las marcas de tiempo (`fecha`) para enriquecer el conjunto de datos.

- **Mes:** Captura variaciones estacionales (`mes`).
- **Día:** Distingue fechas específicas (`dia`).
- **Día de la semana:** Identifica patrones semanales (`dia_semana`).
- **Fin de semana:** Variable binaria para días de descanso (`fin_semana`).

Ventajas:

- Captura efectos estacionales y temporales.
- Mejora el rendimiento de los modelos en datos temporales.

Selección de características

Selección de Características

Selección de características busca identificar las variables más relevantes para predecir el objetivo, mejorando el rendimiento y reduciendo el ruido.

- **Método estadístico:**
  - Usamos `SelectKBest` con pruebas F (*f-test*) para seleccionar las k mejores características.

  Mide la relación lineal entre cada característica y la variable objetivo usanod el F-estadístico

El F-Estadístico

El **F-estadístico** se calcula como:

$$F = \frac{\text{Variación explicada por el modelo}}{\text{Variación no explicada (residual)}}$$

En términos de regresión lineal:

$$F = \frac{\left(\frac{\text{SSR}}{p}\right)}{\left(\frac{\text{SSE}}{n-p-1}\right)}$$

Donde:

- **SSR (Suma de Cuadrados de Regresión):** Variación explicada por el modelo.

$$\text{SSR} = \sum_{i=1}^{n} (\hat{y}_i - \bar{y})^2$$

  ($\hat{y}_i$: Predicciones del modelo, $\bar{y}$: Media de y).
- **SSE (Suma de Cuadrados de los Errores):** Variación no explicada por el modelo (residual).

Interpretación del F-Estadístico

Interpretación:

- **Valor alto de F:**
  - Indica que el modelo explica una proporción significativa de la varianza en los datos.
  - Sugiere que las características tienen una relación significativa con la variable objetivo.
- **Valor bajo de F:**
  - Indica que el modelo no explica bien la variación en los datos.
  - Sugiere que las características pueden no ser relevantes.

- **Método basado en modelos:**
  - Usamos `Random Forest` para calcular la importancia de las características.

Ventajas:

- Reduce la complejidad del modelo.
- Mejora la interpretabilidad y evita el sobreajuste.

Resultados de Selección de Características

- Seleccionar características relevantes mejora el rendimiento y reduce el tiempo de entrenamiento.
- Los datos temporales enriquecidos aportan valor a los modelos predictivos.