



CLASIFICACION CON SVM CON FUNCIÓN RADIAL

Inteligencia Artificial

Nombre:

Pacheco Reyes Kimberlyn

Profe:

Ríos Félix José Mario

Horario:

6:00 – 7:00 PM

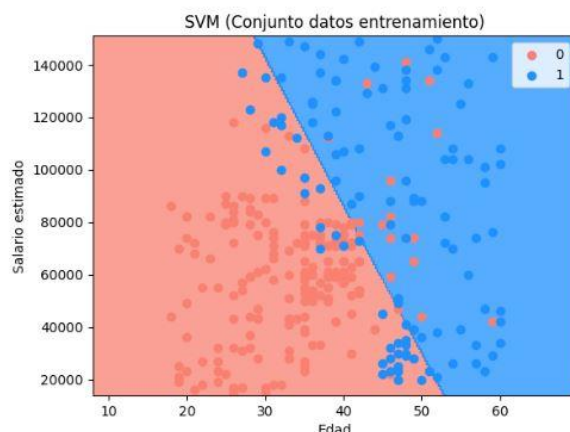
CLASIFICACION CON SVM CON FUNCIÓN RADIAL

Máquinas de Vector Soporte es un algoritmo de clasificación y regresión desarrollado en la década de los 90, dentro del campo de la ciencia computacional. Aunque inicialmente se desarrolló como un método de clasificación binaria, su aplicación se ha extendido a problemas de clasificación múltiple y regresión. SVMs ha resultado ser uno de los mejores clasificadores para un amplio abanico de situaciones, por lo que se considera uno de los referentes dentro del ámbito de aprendizaje estadístico y machine learning.

Las Máquinas de Vector Soporte se fundamentan en el Maximal Margin Classifier, que, a su vez, se basa en el concepto de hiperplano.

Como técnica de clasificación, las SVM son similares a la regresión, pero las SVM ponen el énfasis en el concepto de margen.

El margen es la distancia perpendicular desde un hiperplano separador dado hasta la observación del conjunto de entrenamiento más cercana a él.



TIPOS DE CLASIFICADORES DE SVM

SVM lineal

El clasificador SVM lineal se fundamenta en la búsqueda de un hiperplano que divida el espacio de características en dos regiones, cada una correspondiente a una clase distinta. Su objetivo es maximizar el margen entre los vectores de soporte de ambas clases, garantizando así una separación óptima. Este enfoque resulta particularmente eficiente en problemas donde los datos son linealmente separables o aproximadamente lineales, ofreciendo simplicidad, rapidez en el entrenamiento y facilidad de interpretación. Es ampliamente utilizado en tareas de clasificación de texto y datos tabulares de alta dimensión.

SVM No Lineal

El clasificador SVM no lineal surge como respuesta a los problemas en los que los datos no pueden ser separados mediante un hiperplano recto en su espacio original. Para abordar esta limitación, se emplea el denominado *kernel trick*, una técnica matemática que proyecta los datos a un espacio de mayor dimensión donde la separación lineal resulta posible.

Los SVM no lineales son altamente efectivos en problemas de clasificación con fronteras de decisión irregulares, como el reconocimiento de imágenes, la bioinformática y la detección de anomalías. Su principal desafío radica en la selección adecuada del kernel y de sus parámetros, ya que una configuración incorrecta puede derivar en sobreajuste o en un rendimiento subóptimo.

SVR (Support Vector Regression)

Constituye una extensión del SVM orientada a problemas de regresión. En lugar de separar clases, busca ajustar una función que prediga valores continuos dentro de un margen de tolerancia definido. SVR conserva la filosofía de maximización del margen, adaptada al contexto de regresión.

CREACION DE UN CLASIFICADOR DE SVM

Preparación de los datos

En esta etapa se recopila y organiza el conjunto de datos, asegurando que cada instancia esté correctamente etiquetada. Se aplican técnicas de normalización o estandarización para homogeneizar las escalas de las variables, dado que el SVM es sensible a las magnitudes. Finalmente, se divide el dataset en subconjuntos de entrenamiento y prueba, lo que permite evaluar posteriormente la capacidad de generalización del modelo.

Selección del kernel

El kernel constituye el núcleo del clasificador SVM, pues define la forma de la frontera de decisión. El kernel lineal se emplea cuando los datos son separables mediante un hiperplano recto, mientras que los kernels polinomial, radial (RBF) y sigmoide permiten abordar problemas no lineales al proyectar los datos en espacios de mayor dimensión. La elección del kernel depende de la complejidad y distribución de los datos, siendo un factor determinante en el rendimiento del modelo.

EJEMPLO CON CÓDIGO

SVM lineal

```
from sklearn import datasets
from sklearn.model_selection import train_test_split
from sklearn.svm import SVC
from sklearn.preprocessing import StandardScaler

# Cargar dataset de dígitos
X, y = datasets.load_digits(return_X_y=True)

# Normalizar datos
scaler = StandardScaler()
X = scaler.fit_transform(X)

# Dividir en entrenamiento y prueba
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=42)

# Clasificador SVM lineal
clf_linear = SVC(kernel='linear', C=1)
clf_linear.fit(X_train, y_train)

print("Exactitud (Lineal):", clf_linear.score(X_test, y_test))
```