

SVR - Support Vector Machine para Regresión

SVR (R)

Los SVR utilizan una función (denominada función de Núcleo o *kernel*) para mapear los puntos del conjunto de datos disponibles a otra dimensión donde estos sean linealmente separables.

Si imaginamos un espacio en 2D con puntos de 2 clases distintas, la función de núcleo podría levantar a estos puntos a un espacio en 3D donde estos puntos de distintas clases podrían separarse por un plano.

Escenario del problema

Vamos a contratar un nuevo empleado. Nos ha dicho que en su anterior empresa fue Manager Regional durante 2 años y que cobraba 170.000€ al año. Queremos determinar hasta que punto nos dice la verdad para poder negociar con él el salario que queremos ofrecerle en su nuevo puesto.

¡Vamos a ello!

```
# 1. Importar librerías
```

```
library(caTools)
library(ggplot2)
library(e1071)
```

```
# 2. Importar datos
```

```
datos <- read.csv('../Datos/4.1.Salarios2.csv')
datas = datos[2:3] # Eliminamos la columna del título del puesto y nos quedamos con el nivel
```

```
# 3. Separar en Entrenamiento y Validación
```

Recordatorio: no hacemos división de conjuntos porque tenemos muy pocos datos y nuestra intención es hacer una predicción lo más precisa posible.

```
# 4. Construir el Modelo
```

```
regresor <- e1071::svm(formula = Salario ~ Nivel,
                       data = datos,
                       type = 'eps-regression')
```

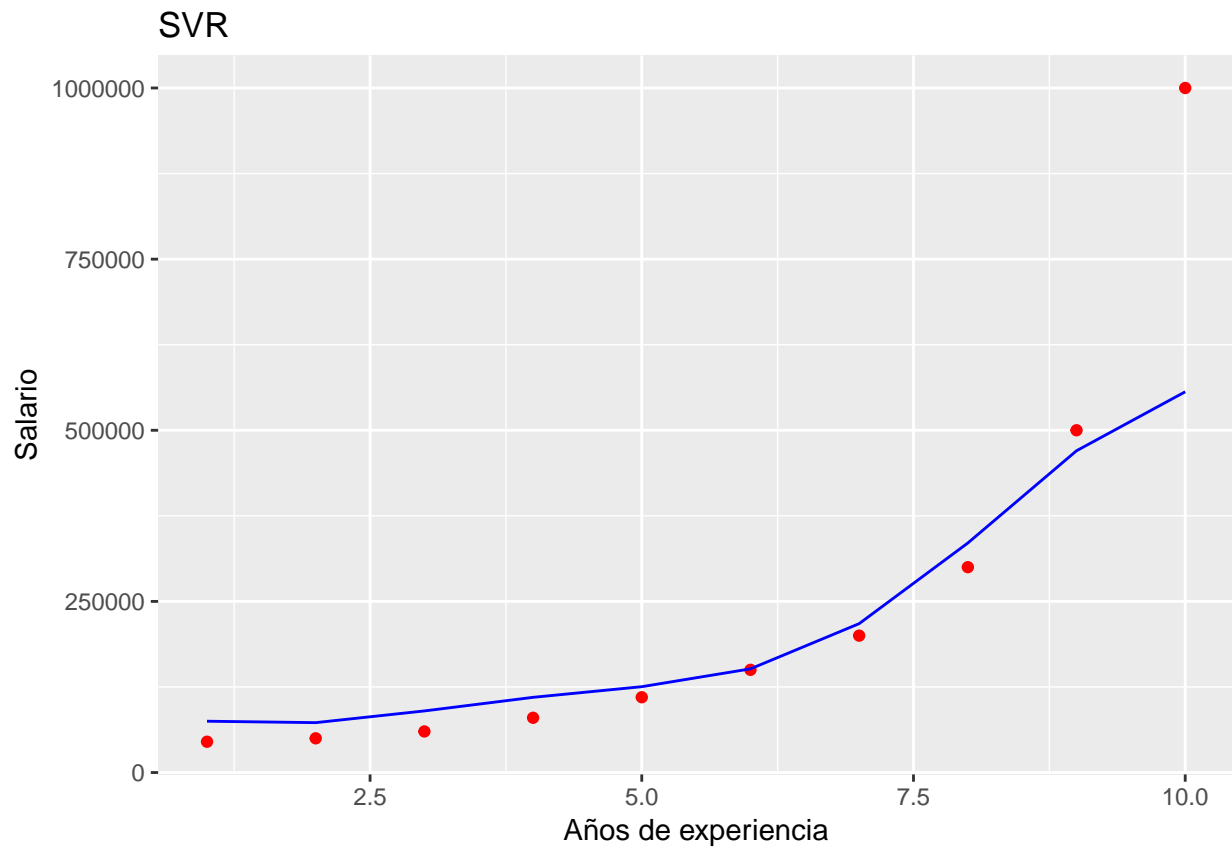
```
# 5. Hacer las predicciones para el conjunto de Validación
```

```
y_fit <- predict(regresor, newdata = datos)
```

No estamos prediciendo en este ejemplo, sino determinando los parámetros para que el modelo se **ajuste** lo mejor posible a los datos del conjunto de entrenamiento (que constituyere todos los datos)

```
# 6. Echemos un vistazo a la pinta que tienen las predicciones
```

```
ggplot() +
  geom_point(aes(datos$Nivel, datos$Salario), colour='red') +
  geom_line(aes(datos$Nivel, y_fit), colour='blue') +
  ggtitle('SVR') +
  xlab('Años de experiencia') +
  ylab('Salario')
```



```
# 7. Calcular el error
library(Metrics)
y_real <- datos$Salario
RMSE <- rmse(y_real, y_fit)
print(RMSE)
```

```
## [1] 142425.4
```