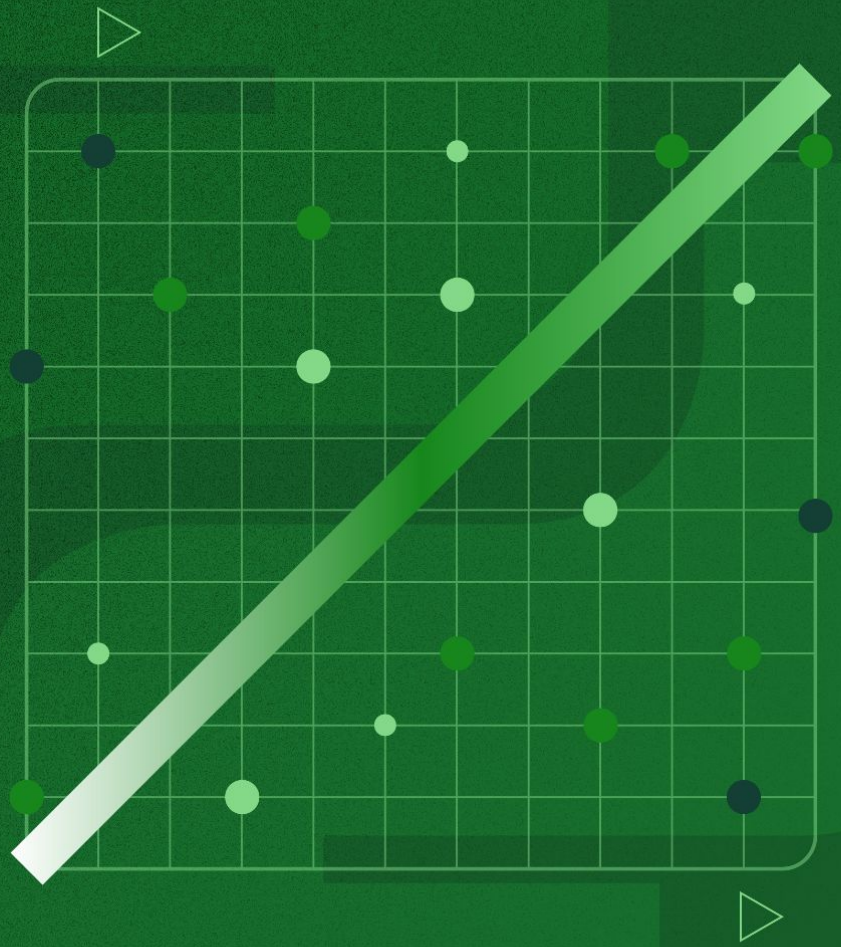
Curso de

Regresión Lineal con Python y scikit-learn

Luis Laris



¿Quién es Luis Laris?

- Actuario.
- Msc. en desarrollo de juegos con enfoque a IA.
- Trabajado como científico de datos para consultoras en ciencia de datos, KADK, Ubisoft, ITU.
- Student Team Lead en Platzi Master.

Requisitos previos

- Matemáticas para inteligencia artificial.
- Análisis exploratorio de datos con Python y Pandas.
- Visualización de datos con Matplotlib y Seaborn.
- Fundamentos de inteligencia artificial.

```
import seaborn as sns
sns.set(style='whitegrid', context='notebook')
sns.pairplot(df[cols], height=2.5)
sns.heatmap(cm,cbar=True,
                    annot=True,
                    square=True,
                    fmt='.2f',
                    annot_kws={'size':15},
                    yticklabels=cols,
                    xticklabels=cols)
```

Análisis de datos para tu primera regresión lineal

Entrenando un modelo de regresión lineal con scikit-learn

¿Qué es la regresión lineal?

Regresión lineal

1. Genera una línea de datos que mejor se ajusta a los datos originales.
2. Forma parte de análisis supervisado.
3. W_0 es el intercepto y W_1 es la pendiente.

Función de pérdida y optimización

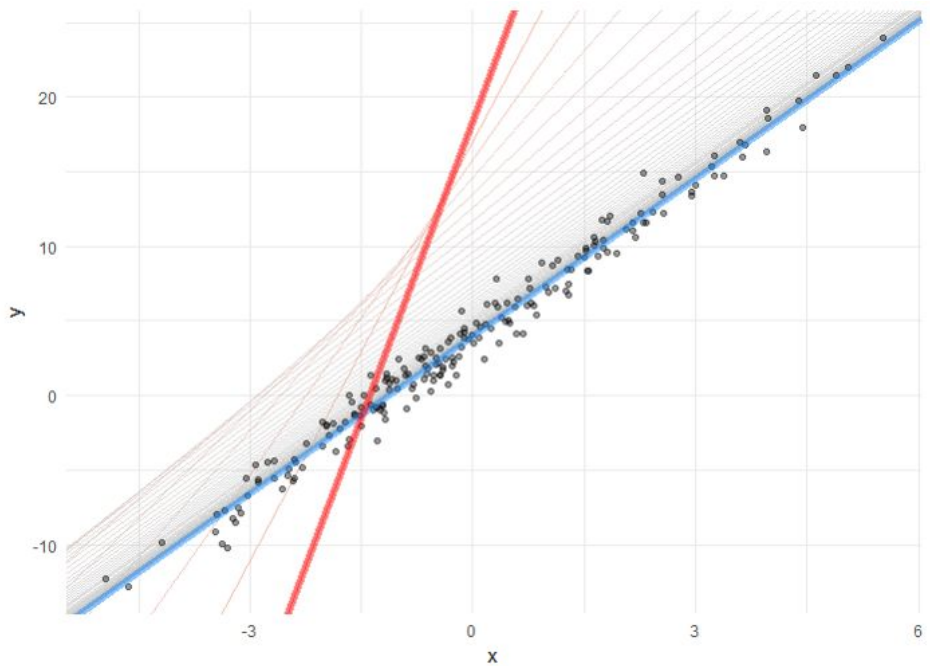
Análisis usando mínimos cuadrados

Pasos del algoritmo

Un algoritmo de ML siempre pasa por un loop:

1. Se ajusta el modelo.
2. Se comparan resultados con los reales.
3. Se ajustan pesos en el modelo.
4. Regreso a paso inicial si no se converge.

¿Cómo se ve el paso a paso?



¿Qué modificamos?

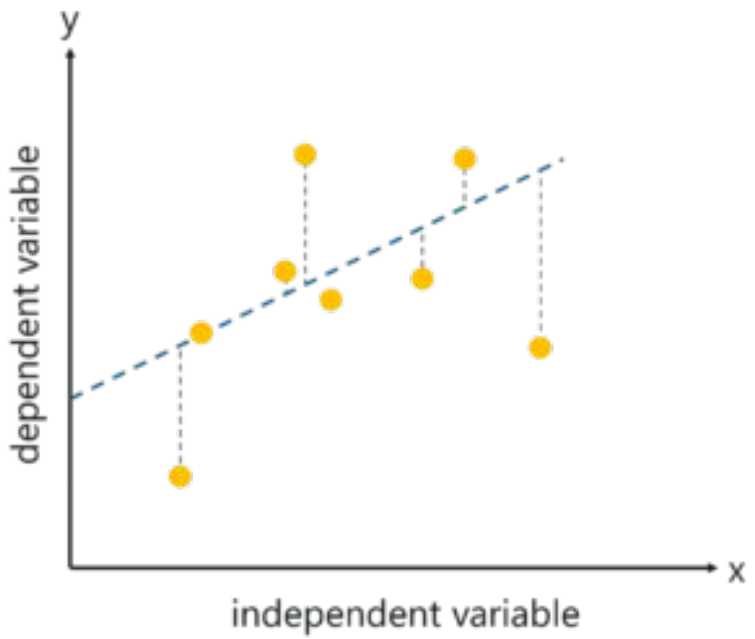
$$\hat{y} = w_0 + w_1 X$$

Para ejecutar este algoritmo

- Función de pérdida:
  método de mínimos cuadrados.

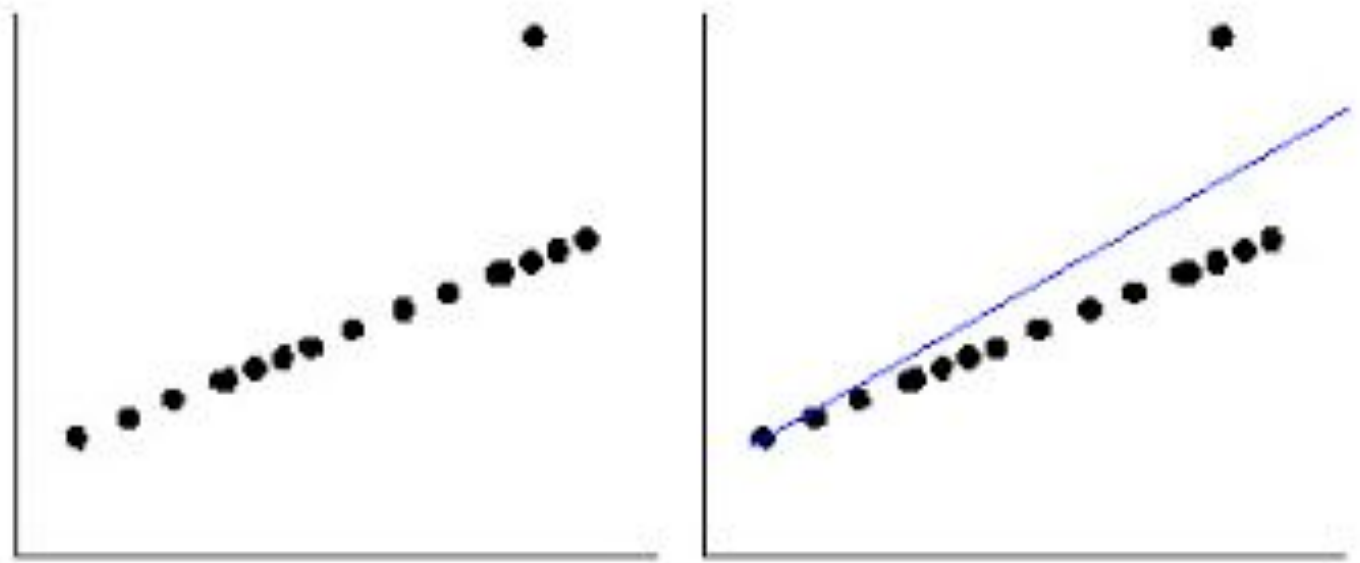
- Algoritmo de optimización:
  el más común es el de **descenso del gradiente**.

Función de pérdida



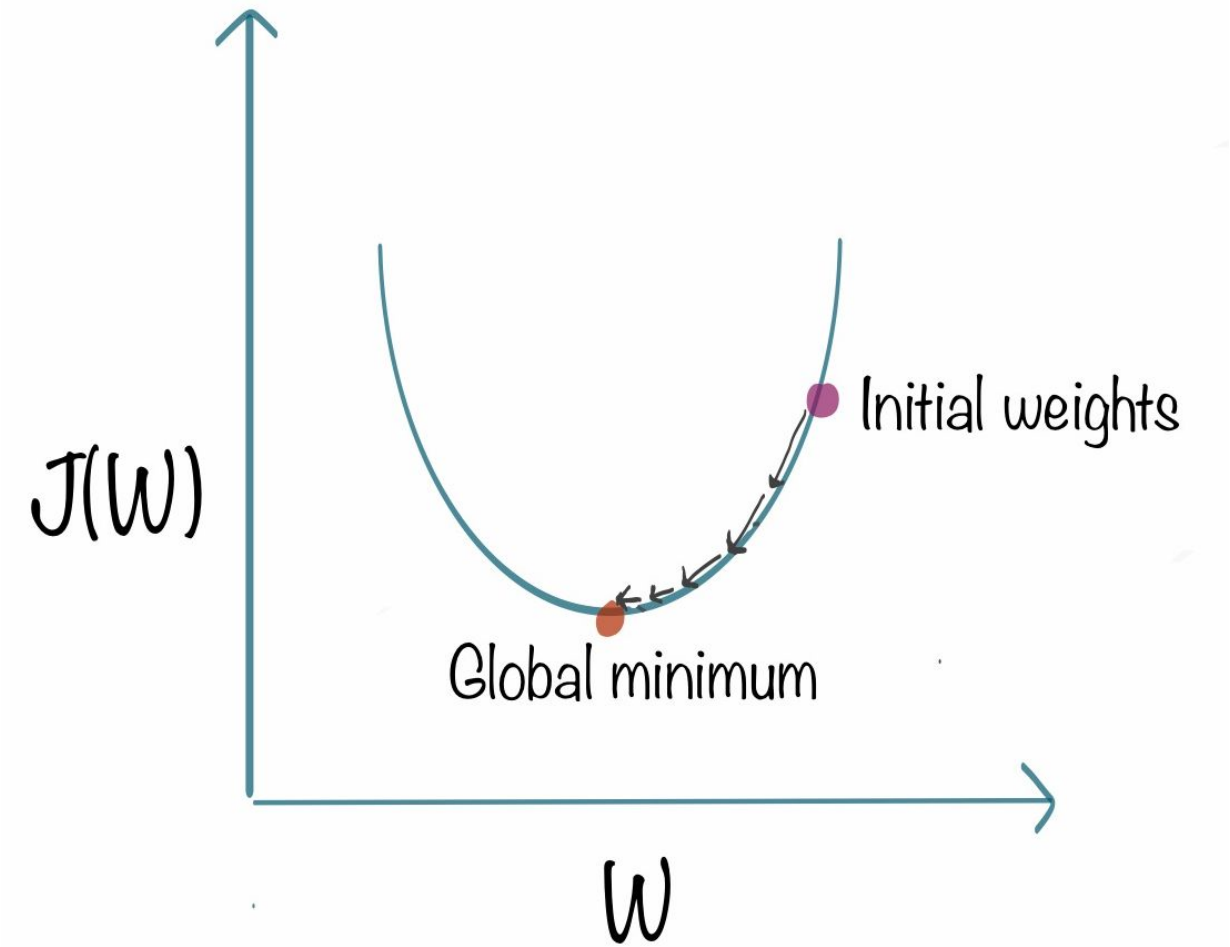
$$J = \frac{1}{N} \sum_{1}^{N} (y_i - \hat{y}_i)^2$$

Solo un problema



Valores atípicos (outliers)

Algoritmo de Optimización



Evaluando el modelo

R^2 y MSE

Empezar con MSE

- **MSE (Mean Square Error)** se usa para evaluar función de pérdida.
- Es bueno revisar cuál fue este mínimo hallado por el modelo.

MSE

$$MSE = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2$$

A valores grandes, peor se ajusta el modelo.
*problema de dimensionalidad

Coeficiente de determinación R^2

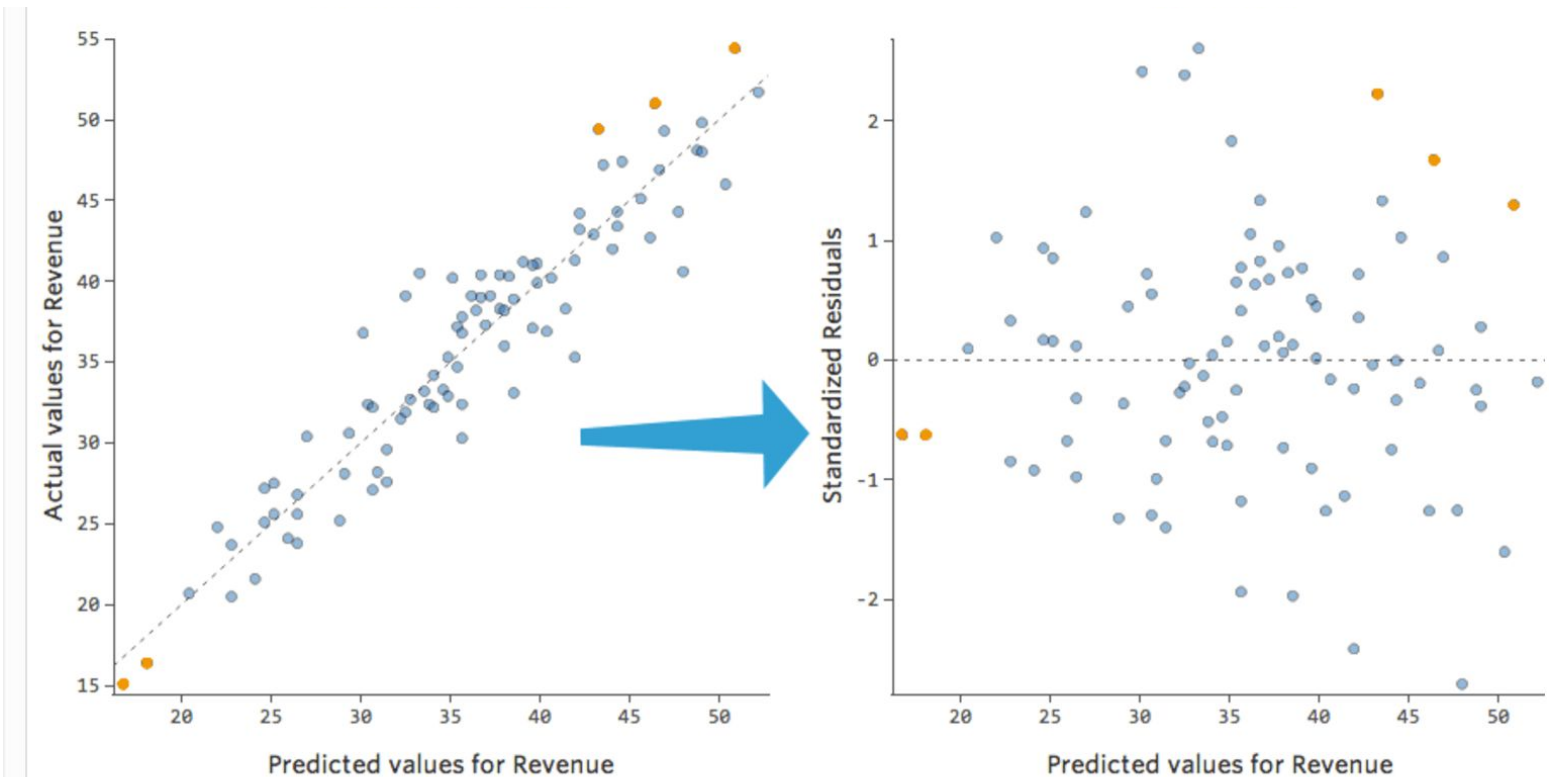
$$R^2 = 1 - \frac{\sum_{i=1}^{N}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{N}(y_i - \bar{y})^2}$$

Otras métricas importantes a considerar

- R - Ajustada.
- Error máximo.
- Error absoluto promedio (MAE).
- Mediana de los errores absoluto.
- Raíz del promedio de los errores cuadrados (RMSE).
- Varianza explicada.

Pero no solo eso

Los residuales



Regresión lineal multivariable

Lo mismo,
pero con más variables

$$\hat{y} = w_0 + w_1 * x_1 + ... + w_n * x_n$$

Análisis de regresión multivariable

Cuándo utilizar un modelo de regresión lineal

Las grandes preguntas

- ¿Tengo que predecir sobre una variable numérica?
- ¿Las variables independientes con las que cuento son primordialmente numéricas?
- No cuento con una gran cantidad de variables y / o variables categóricas con muchos niveles.

Recomendaciones

- Si tienes que predecir variables numéricas empieza con regresión lineal, si no funciona, salta a otros modelos.
- Reduce las variables lo más que puedas.
- ¡Cuidado con la multicolinealidad!
- No predigas fuera del dominio de la variable independiente.

Regresión lineal para predecir los gastos médicos de pacientes

Proyecto práctico

Nuestros datos

Utilizando como inicio datos del US Census Bureau se recabaron datos de gastos en seguros de datos médicos de varios pacientes.

Variable	Tipo	Descripción
Edad	numérica	La edad del asegurado
Sexo	dicotómica	Género binario del asegurado
Índice de masa corporal	numérica	Índice de masa corporal del paciente
Hijos	numérica entera	Cantidad de hijos del paciente
Fumador	booleana	Si el paciente es fumador o no fumador
Región	categórica	Región en la que vive el paciente
Cargos	numérica	La cantidad que pago de seguro el paciente

Exploración y preparación de datos

Proyecto práctico

Análisis de correlación de los datos

Proyecto práctico

Entrenamiento del modelo

Proyecto práctico

Evaluando el modelo

Proyecto práctico

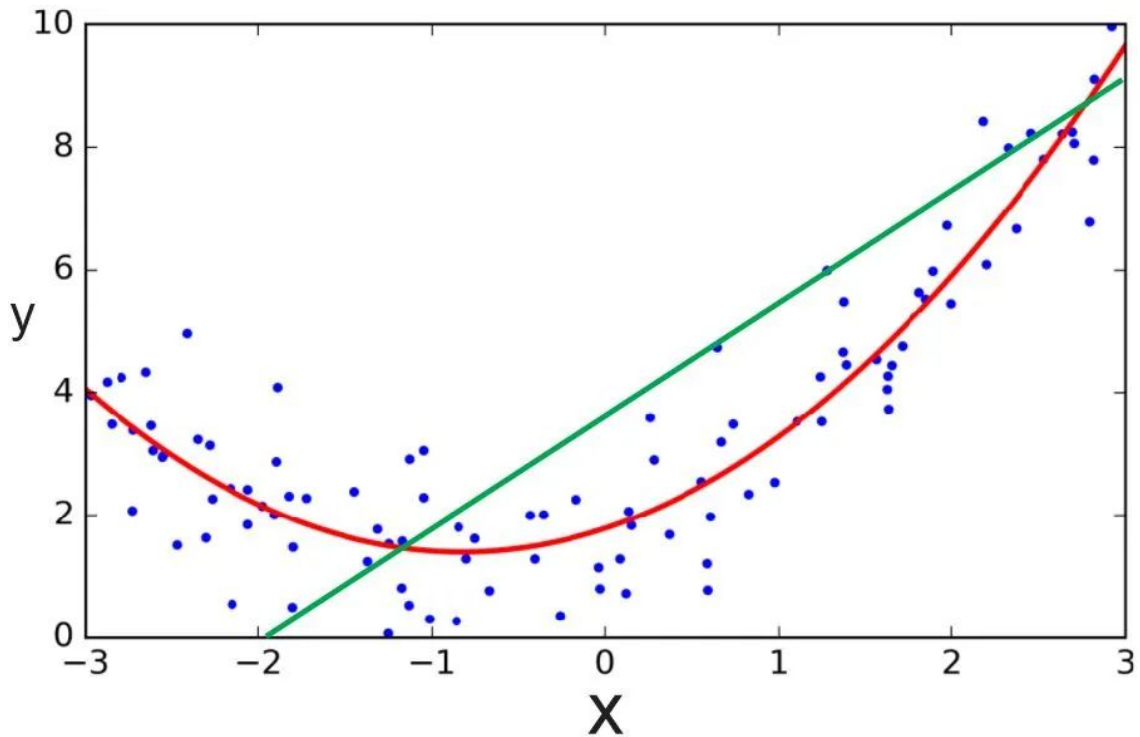
```python
import sklearn.metrics as metrics

# metricas de regresión
explained_variance=metrics.explained_variance_score(y_true, y_pred)
mean_absolute_error=metrics.mean_absolute_error(y_true, y_pred)
mse=metrics.mean_squared_error(y_true, y_pred)
median_absolute_error=metrics.median_absolute_error(y_true, y_pred)
r2=metrics.r2_score(y_true, y_pred)
```

Mejorando el modelo

Proyecto práctico

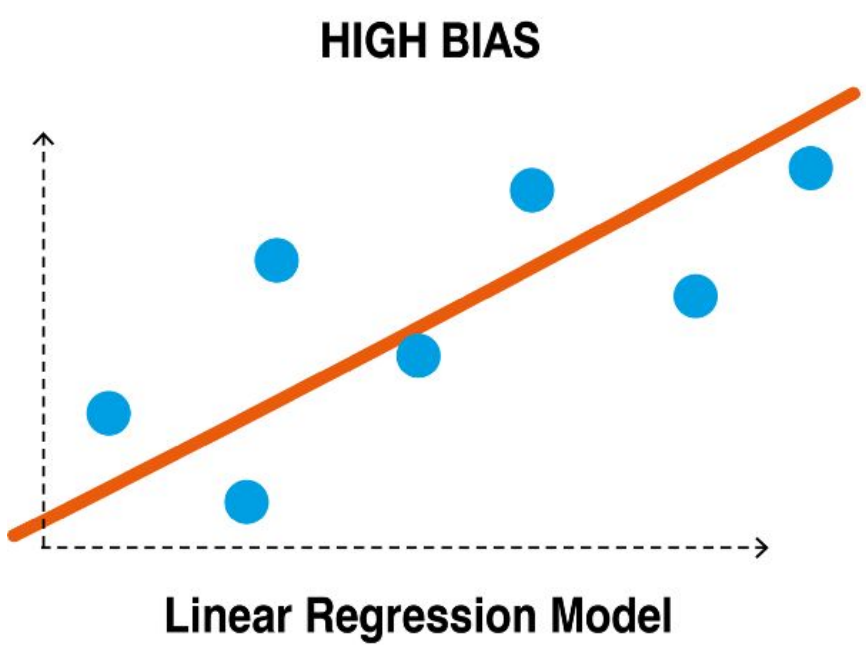
¿Qué hay más allá de la linealidad?

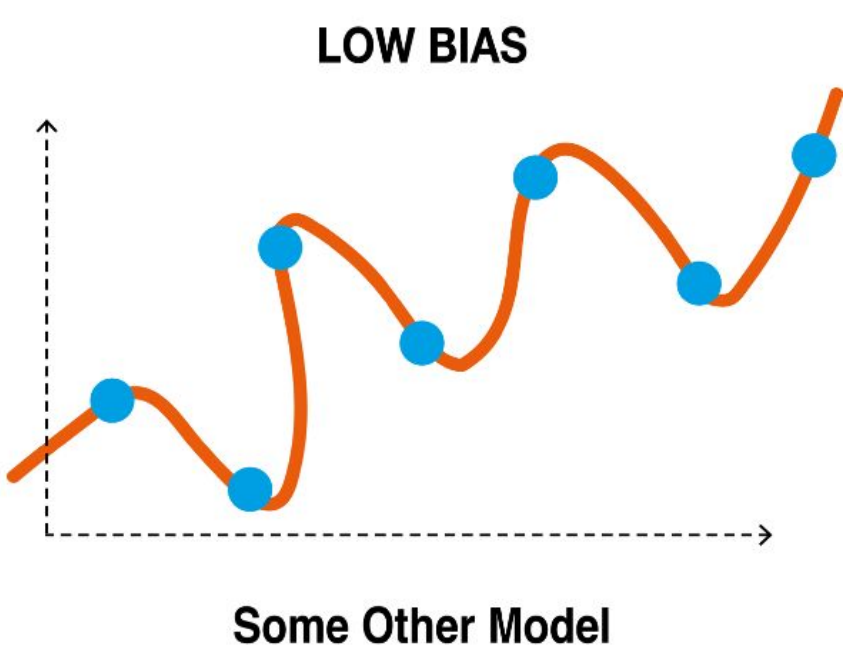
Modelos polinomiales



$$\hat{y} = w_0 + w_1 x + w_2 x^2 + \ldots + w_n x^n$$

Sesgo y varianza

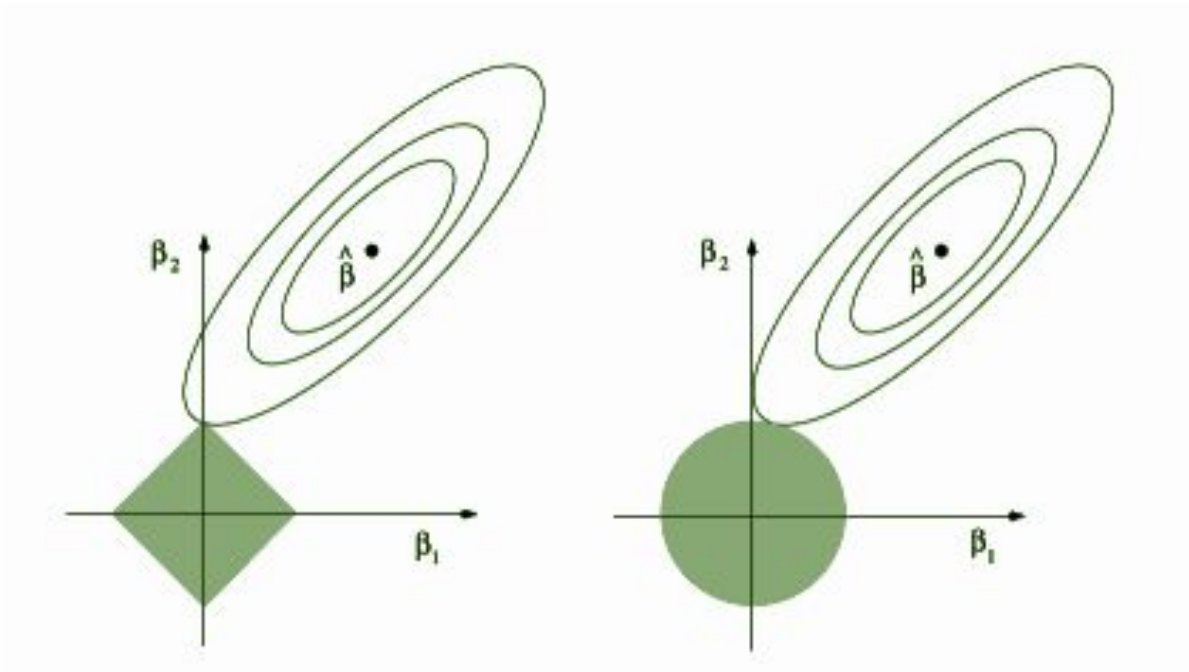




Regularización

- Se cambia la función de pérdida para que tenga una penalización.
- Reduce la varianza mientras aumenta el sesgo.

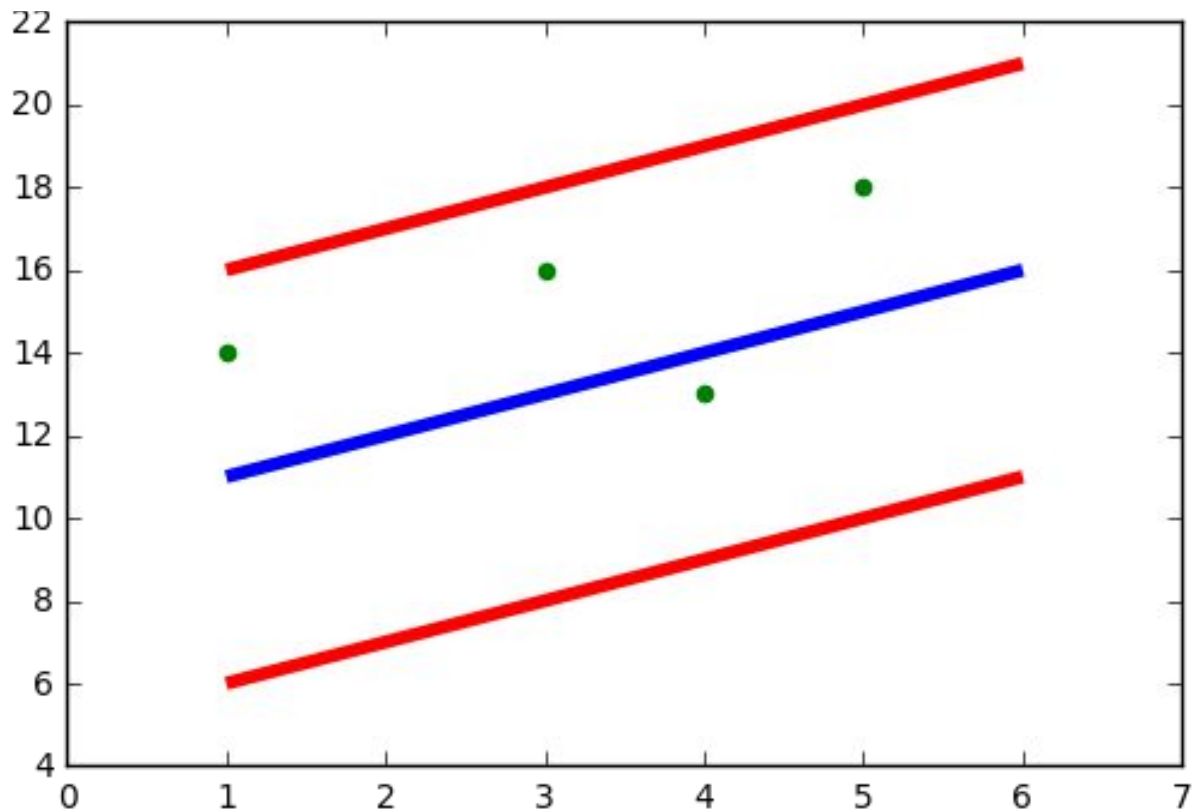
Regularización



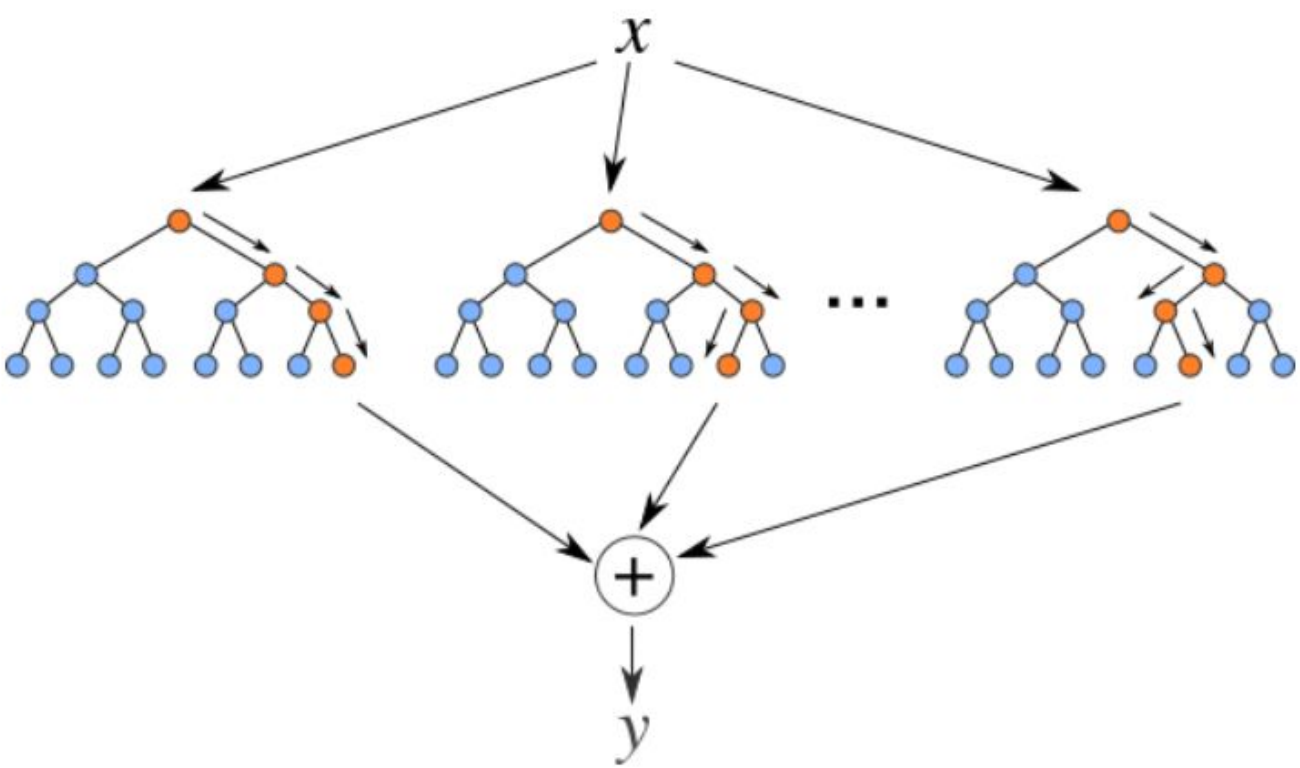
$$J = \frac{1}{N}\sum_{i=1}^{N}(y_i + \hat{y}_i) - \lambda \sum_{j} Reg(w_j)$$

Siguientes pasos en modelos de inteligencia artificial

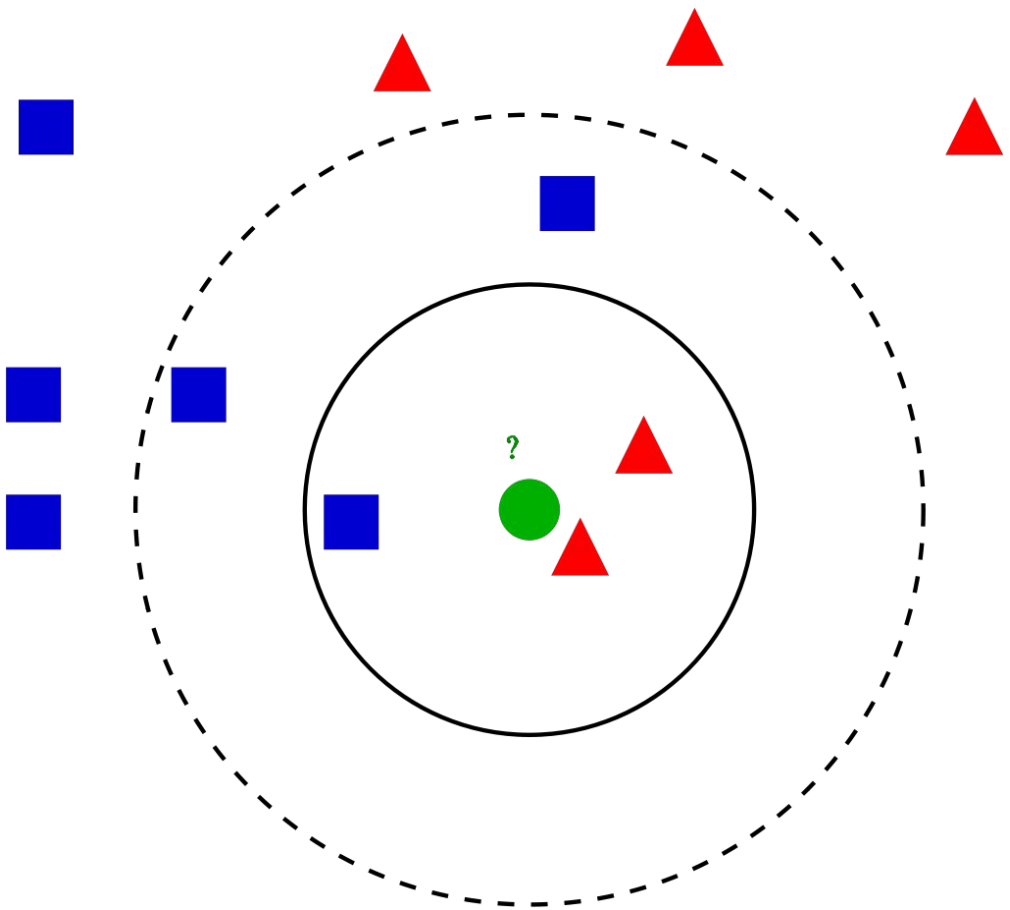
Vectores de soporte de regresión



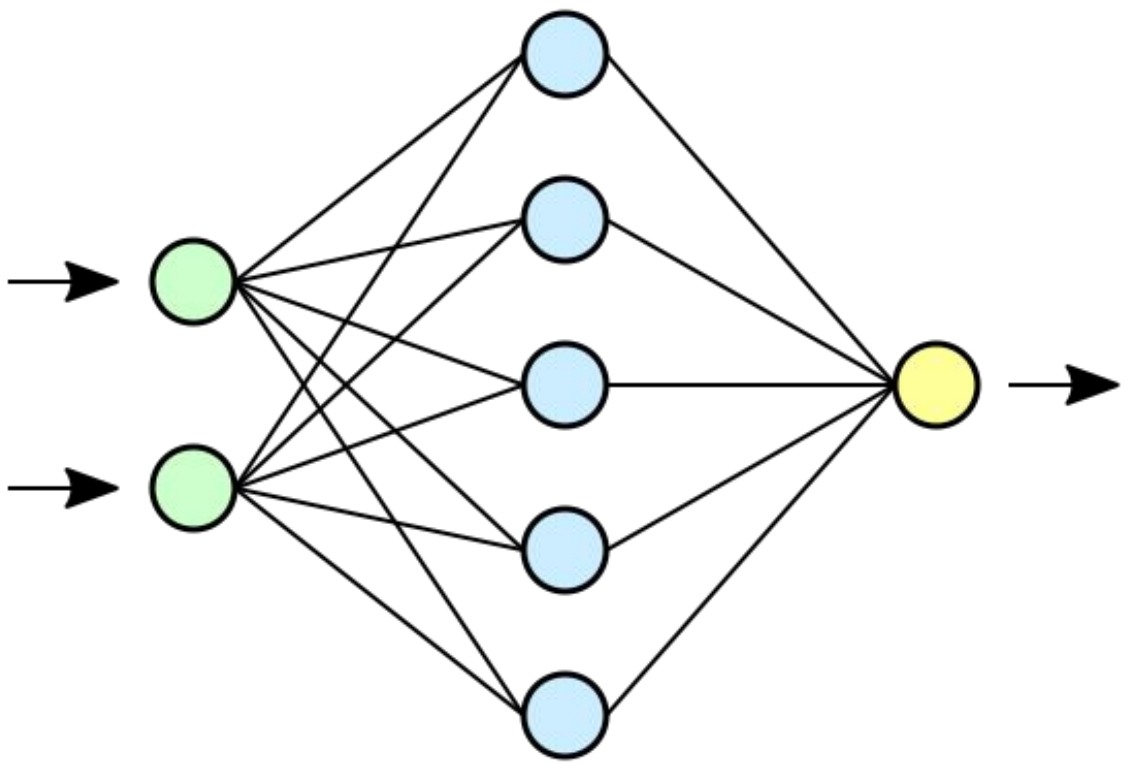
Bosques aleatorios



K-vecinos más cercanos



Redes neuronales



Y no solo ahí

- Vimos análisis supervisado para problemas de regresión.
- Existe análisis supervisado para la clasificación y el análisis no supervisado para clustering y reducción de dimensionalidad.