Linear Regression

محاضرة 3: المبادئ والتطبيقات

مقدمة في Supervised Learning والنمذجة الرياضية

فكرة Linear Regression

الهدف والتعريف

Linear Regression هو أبسط نموذج في Supervised Learning.

الهدف: توقع قيمة y (مثلاً: سعر بيت) بناءً على features معينة (مثل المساحة).

الفكرة: نفترض أن العلاقة بين المدخلات والمخرجات علاقة خطية تقريباً.

المعادلة الرياضية

معادلة النموذج بمتغير واحد:

$$\hat{y} = wx + b$$

- x: الـ feature (المدخل).
- w: الـ weight (الميل).
- b: الـ bias (نقطة التقاطع).

الميل (Slope) و معدل التغير

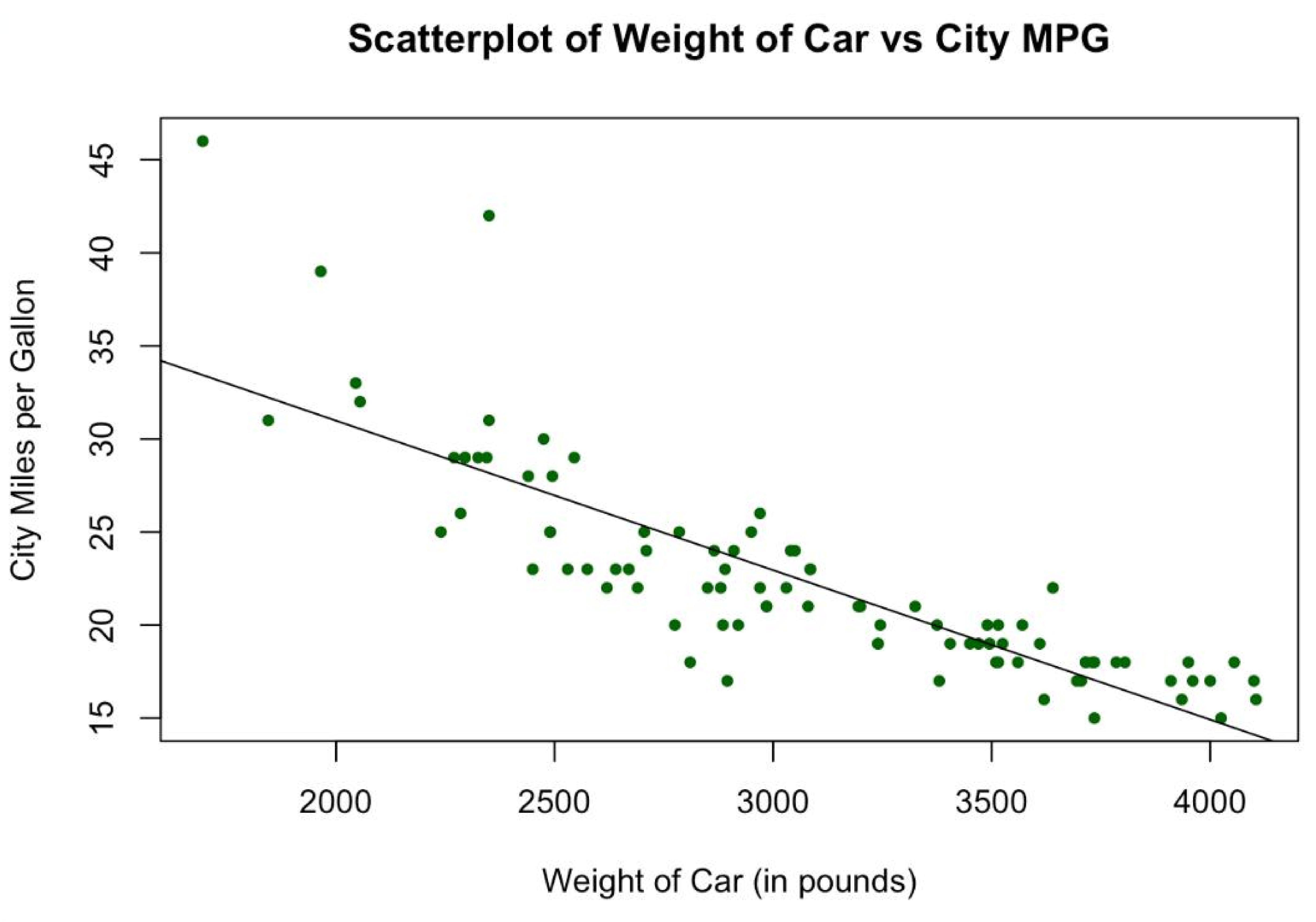
تفسير الـ Slope (w)

الـ Slope يعبر عن معدل تغير y بالنسبة لـ x.

$$w = \frac{\Delta \hat{y}}{\Delta x}$$

الأهمية في الـ Business:

- إذا كان w كبيراً: الـ feature يؤثر بقوة في السعر.
- يعطي Interpretability عالية للنموذج أمام الـ Stakeholders.



تمثيل بياني يوضح خط الانحدار وعلاقته بالبيانات

المفاهيم الأساسية: Error, Loss & Cost

Error (Residual)

الفرق بين القيمة الحقيقية والمتوقعة لنقطة واحدة.

$$e = \hat{y} - y$$

Loss Function

حساب الخطأ لنقطة واحدة (غالبًا نستخدم التربيع لمعاقبة الأخطاء الكبيرة).

$$L = (\hat{y} - y)^2$$

Cost Function (J)

متوسط الخطأ لكل البيانات (MSE). الهدف هو تقليل هذه القيمة.

$$J = \frac{1}{2m} \sum (\hat{y} - y)^2$$

اشتقاق الـ Gradient

الهدف من الاشتقاق

نحتاج معرفة اتجاه التغيير المطلوب في w و b لتقليل الخطأ J.

نستخدم قاعدة السلسلة (Chain Rule) لاشتقاق دالة التكلفة بالنسبة للمتغيرات.

يتم استخدام هذه المشتقات (Gradients) في خوارزمية Gradient Descent.

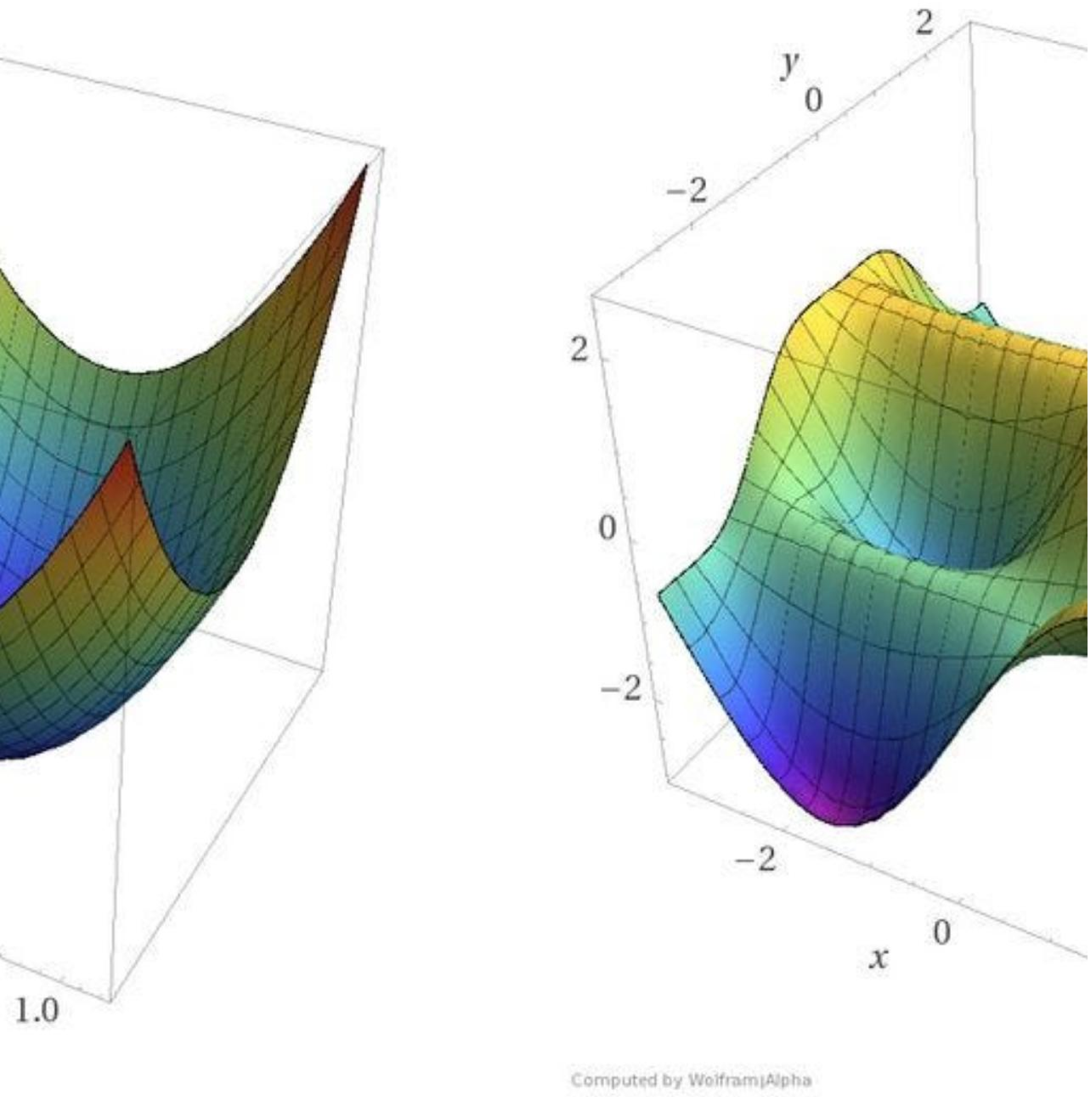
نتائج الاشتقاق (Gradients)

مشتقة بالنسبة لـ w:

$$\frac{\partial J}{\partial w} = \frac{1}{m} \sum_{i=1}^{m} (\hat{y}^{(i)} - y^{(i)}) x^{(i)}$$

مشتقة بالنسبة لـ b:

$$\frac{\partial J}{\partial b} = \frac{1}{m} \sum_{i=1}^{m} (\hat{y}^{(i)} - y^{(i)})$$



Gradient Descent Algorithm

آلية العمل

نبدأ بقيم عشوائية، ثم نتحرك خطوة بخطوة في الاتجاه المعاكس للميل لتقليل التكلفة (Minimize J).

Update Rule

يتم تحديث الأوزان في كل تكرار (Iteration):

$$w = w - \alpha \frac{\partial J}{\partial w}$$

(α) Learning Rate

إذا كان α كبيراً جداً

قد "يقفز" النموذج فوق النقطة الصغرى (Minimum) ولا يستقر أبدًا.

يؤدي ذلك إلى مشكلة **Divergence** (الابتعاد عن الحل الأمثل).

النتيجة: النموذج لا يتعلم.

إذا كان α صغيراً جداً

الخطوات تكون دقيقة ولكن بطيئة للغاية.

يستغرق التدريب وقتًا طويلاً جدًا للوصول للحل (Slow Convergence).

النتيجة: تكلفة حسابية ووقت ضائع.

أنواع Gradient Descent

Batch GD

يستخدم **كل الداتا** في كل خطوة.

✓ مستقر

✕ بطيء مع Big Data

Mini-Batch GD

يقسم الداتا إلى **مجموعات صغيرة** (e.g., 32, 64).

✓ الأفضل عملياً

✓ سريع ومستقر

Stochastic GD

يستخدم **عينة واحدة** في كل خطوة.

✓ سريع جداً

✕ غير مستقر (Noisy)

Polynomial Regression

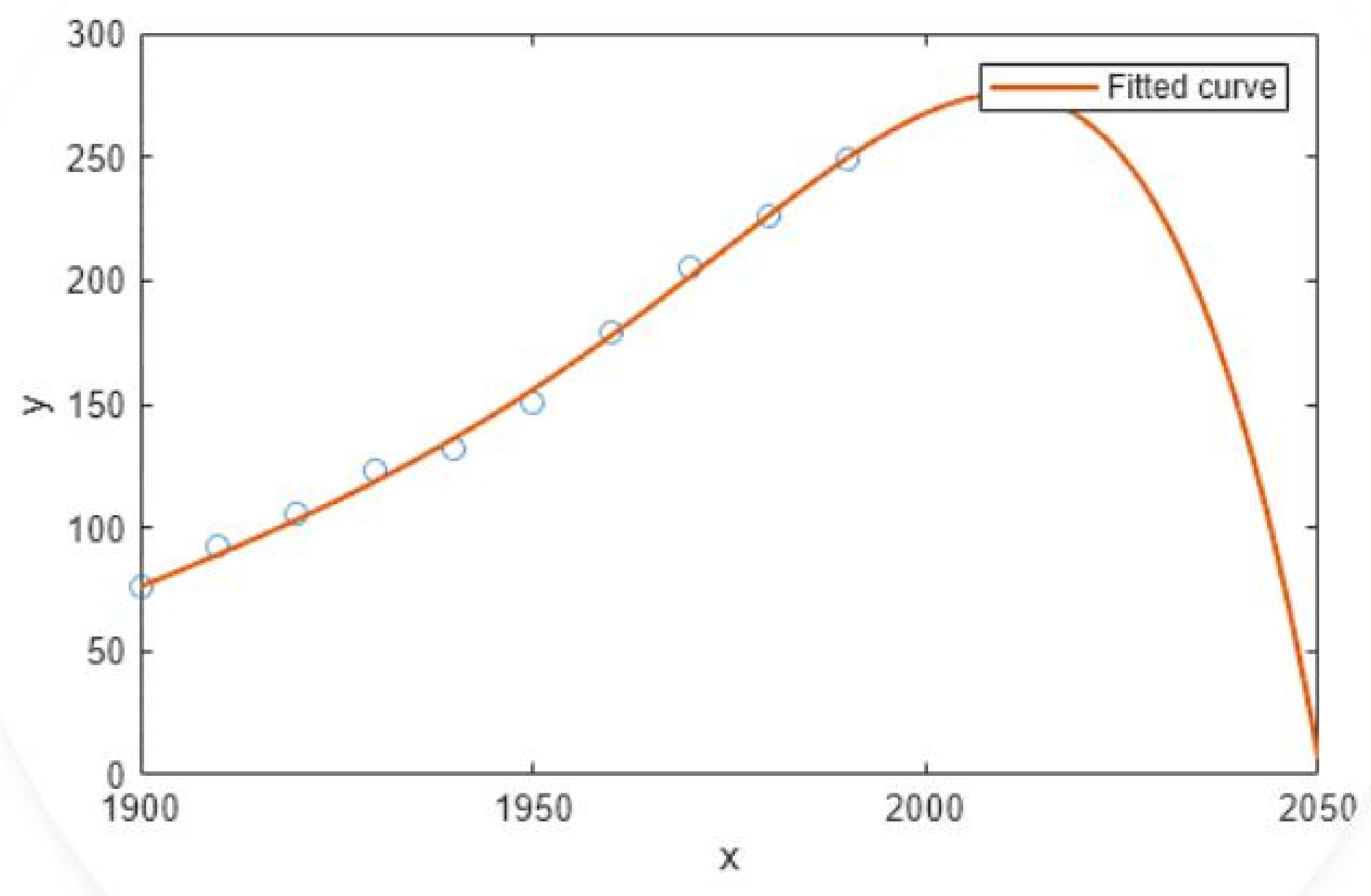
التعامل مع البيانات غير الخطية

عندما تكون العلاقة بين البيانات ليست خطاً مستقيماً (مثلاً منحنى)، لا يكفي النموذج الخطي البسيط.

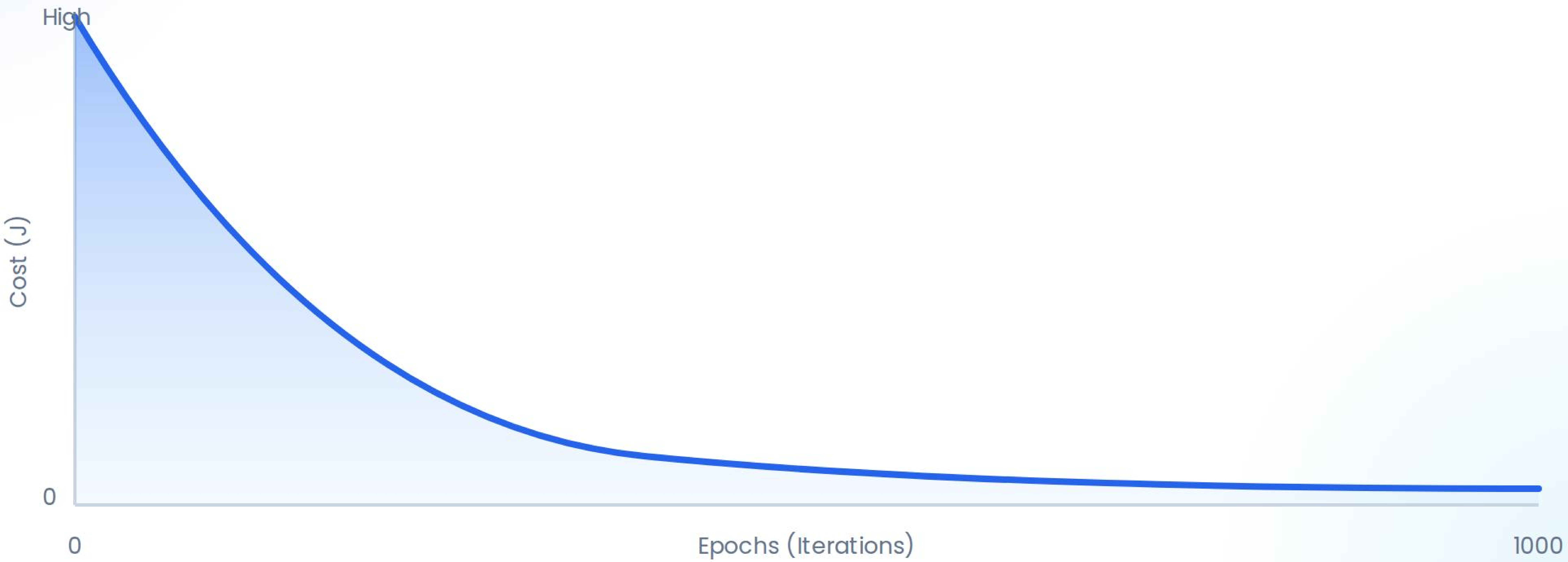
الحل: إضافة Polynomial Features (مثل x^2, x^3).

$$\hat{y} = w_1 x + w_2 x^2 + b$$

* ملاحظة: يجب الحذر من الـ Overfitting عند زيادة الدرجة.



تحليل أداء النموذج (Training Loss)



يوضح الرسم البياني انخفاض دالة التكلفة (Cost Function) مع مرور الوقت (Epochs). هذا الشكل (L-shape) يدل على أن النموذج يتعلم بشكل صحيح ومعدل التعلم مناسب.

ملخص لمهندس الذكاء الاصطناعي

- **Interpretability:** الـ Weights تخبرك بأهمية كل Feature في اتخاذ القرار.
- **Feature Engineering:** إذا كان الـ weight قريب من الصفر، قد يكون الـ Feature غير مفيد.
- **Debugging:** راقب منحنى الـ Loss؛ إذا كان يزداد بدلاً من النقصان، قم بتقليل الـ Learning Rate.
- **Efficiency:** استخدم Mini-Batch GD عند التعامل مع البيانات الكبيرة لتحقيق توازن بين السرعة والاستقرار.

أسئلة؟

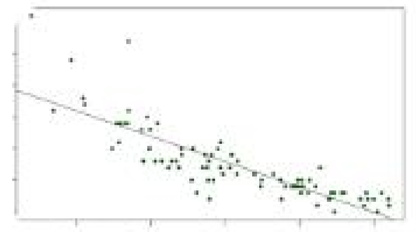
شكراً لاستماعكم

Lecture 3 | Linear Regression

Image Sources

https://bookdown.org/dli/rguide/R-Manual_files/figure-html/unnamed-chunk-181-1.png

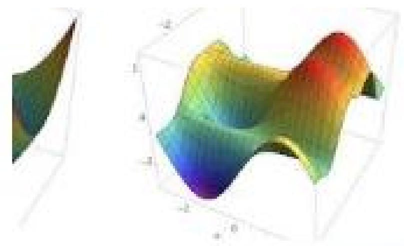
Source: bookdown.org



---

https://miro.medium.com/1*tr43WwL0B3oRKG5FqzlpNw.jpeg

Source: medium.com



---

https://de.mathworks.com/help/examples/curvefit/win64/FitPolynomialExample_06.png

Source: de.mathworks.com