LeanML: Formally Verified Linear Regression

Alisson Matheus Silva

April 10, 2026

Outline

Linear Regression

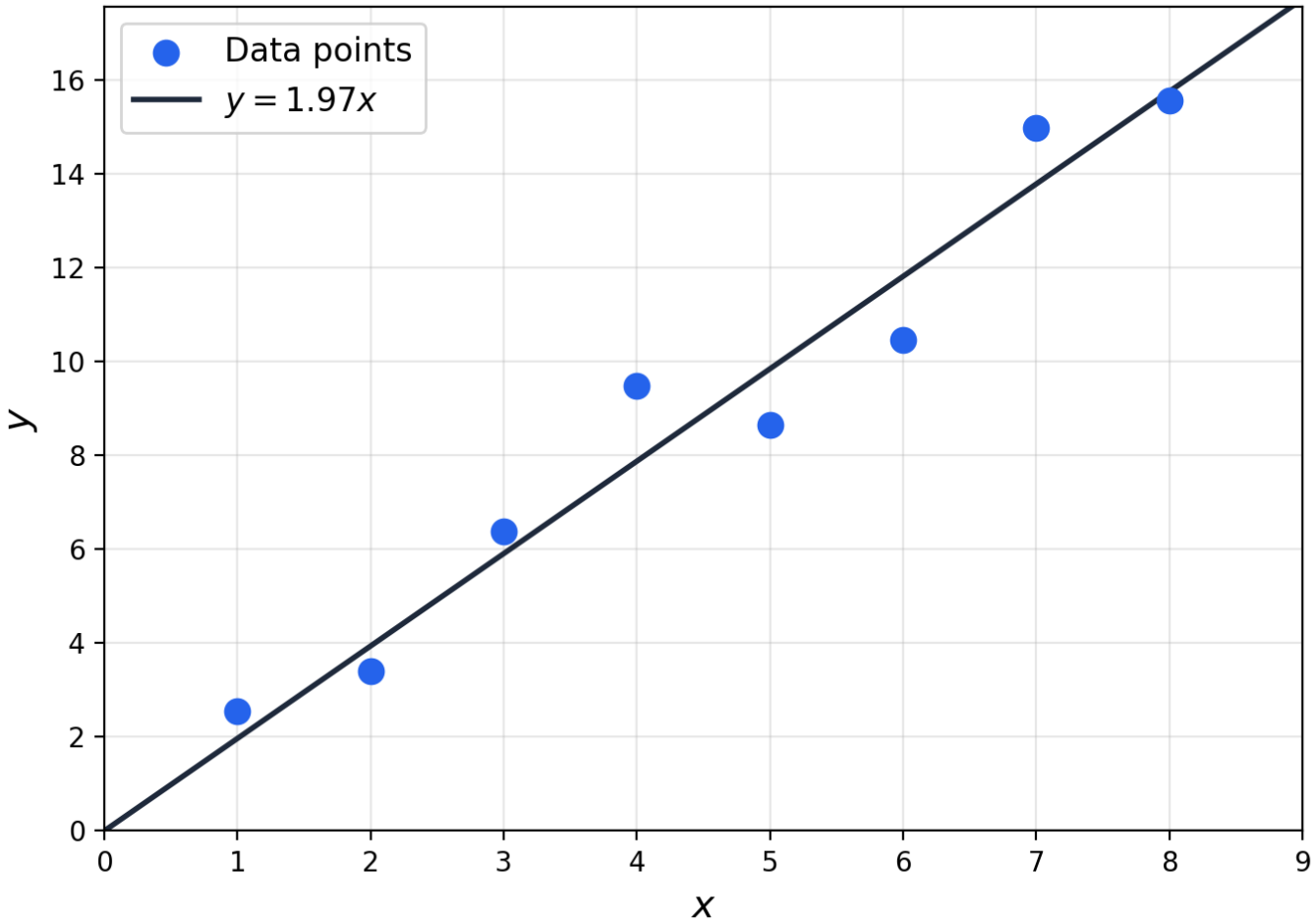

Figure: Linear regression through the origin.

Linear Regression

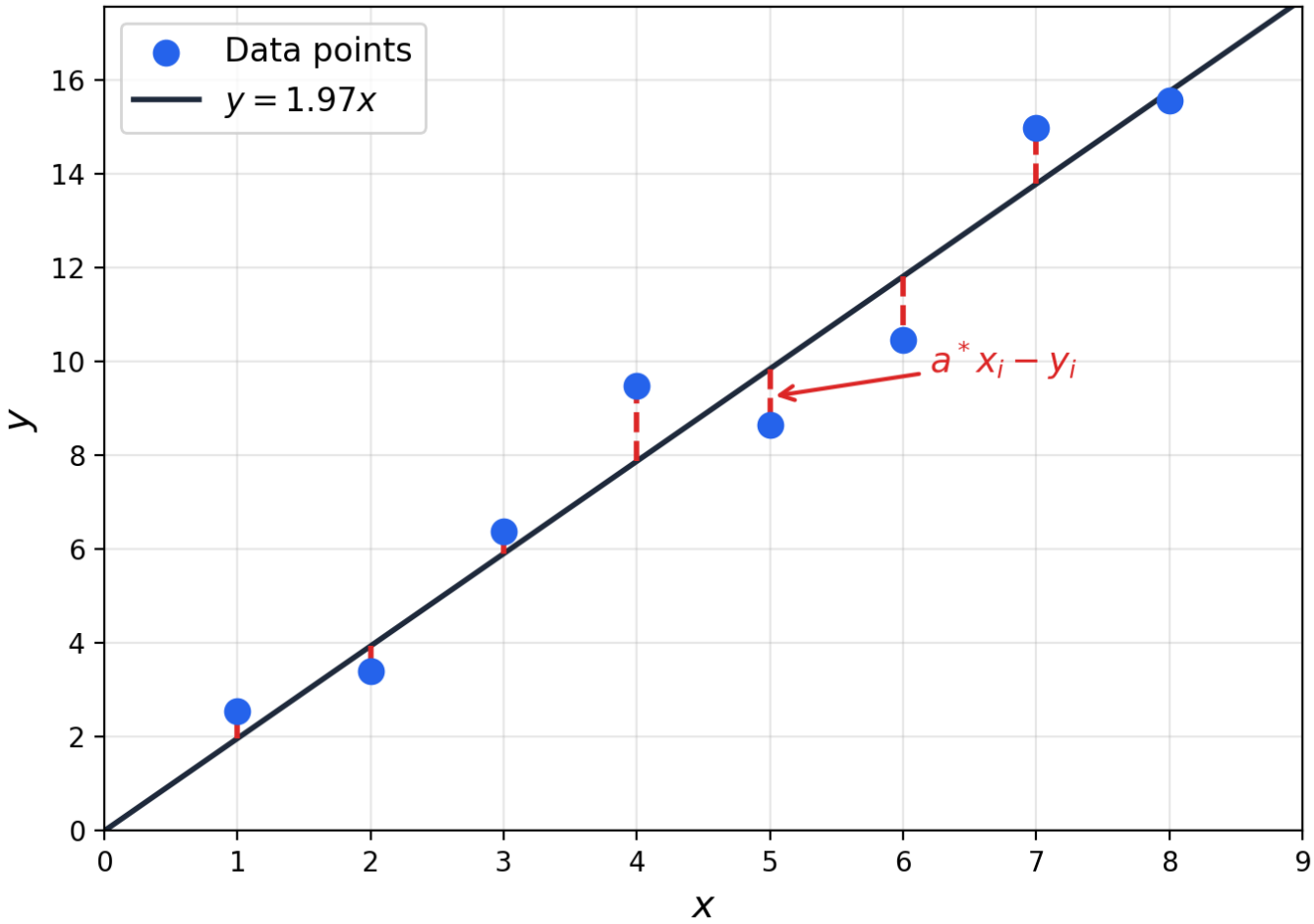

Figure: Linear regression through the origin.

Definitions

- We define the following quantities:
  - Data: n points $(x_1, y_1), \ldots, (x_n, y_n)$ with $x_i, y_i \in \mathbb{R}$
  - Sum of squared inputs: $S_{xx} = \sum_{i=1}^{n} x_i^2$
  - Cross sum: $S_{xy} = \sum_{i=1}^{n} x_i\, y_i$
  - OLS estimator: $a^* = \dfrac{S_{xy}}{S_{xx}}$
  - Squared loss: $L(a) = \sum_{i=1}^{n} (a\, x_i - y_i)^2$

Proof Strategy

- We claim that $a^* = S_{xy}/S_{xx}$ minimizes the squared loss. That is, no other slope can do better:
  - For all $a \in \mathbb{R}$:
    $$L(a^*) \leq L(a)$$
  - The proof relies on decomposing the loss as:
    $$L(a) = L(a^*) + S_{xx}(a - a^*)^2$$
  - This decomposition requires showing that the cross term in $L(a)$ vanishes at a^* and using that S_{xx} is non-negative such that $L(a^*)$ is the only possible minimum.

Lemma Progression

1. Sxx_nonneg: $S_{xx} = \sum x_i^2 \geq 0$
   A sum of squares is nonneg.

2. crossTerm_eq_zero: $\sum x_i(a^* x_i - y_i) = 0$
   The normal equation holds at a^*. This is what kills the middle term in the expansion.

3. loss_decomp: $L(a) = L(a^*) + S_{xx}(a - a^*)^2$
   Split each residual, expand, and use (2) to eliminate the cross term.

4. aStar_minimizes: $\forall\, a,\ L(a^*) \leq L(a)$
   Follows from (3) and (1): the penalty term is nonneg, so a^* is optimal.

Lemma 1: Sxx_nonneg

- The proof proceeds by induction on the finite set:

```
lemma Sxx_nonneg (x : Fin n → ℝ) : 0 ≤ Sxx x := by
  classical
  unfold Sxx
  refine Finset.induction_on (s := (Finset.univ : Finset (Fin n))) ?base ?step
  · simp
  · intro a s ha hs
    have hxa : 0 ≤ (x a) ^ 2 := sq_nonneg (x a)
    simpa [Finset.sum_insert, ha] using add_nonneg hxa hs
```

Lemma 2: crossTerm_eq_zero

- We need to show that the cross term vanishes at a^*:

$$\sum_{i=1}^{n} x_i(a^* x_i - y_i) = 0$$

- Expand and factor:

$$\sum x_i(a^* x_i - y_i) = a^* \sum x_i^2 - \sum x_i y_i = a^* \cdot S_{xx} - S_{xy}$$

- Substitute $a^* = S_{xy}/S_{xx}$:

$$\frac{S_{xy}}{S_{xx}} \cdot S_{xx} - S_{xy} = S_{xy} - S_{xy} = 0$$

- The sum uses `Finset.sum_sub_distrib` and `Finset.mul_sum` to factor, then `field_simp` clears the denominator and `ring` finishes.

Lemma 3: loss_decomp

- Split each residual by adding and subtracting a^*:

$$ax_i - y_i = \underbrace{(a - a^*)x_i}_{\text{perturbation}} + \underbrace{(a^* x_i - y_i)}_{\text{residual at } a^*}$$

- Square both sides and sum over i:

$$\sum (ax_i - y_i)^2 = \sum (a - a^*)^2 x_i^2 + \sum (a^* x_i - y_i)^2 + 2(a - a^*) \sum x_i (a^* x_i - y_i)$$

- The third term is zero by Lemma 2, so we are left with:

$$L(a) = L(a^*) + S_{xx}(a - a^*)^2$$

Lemma 3: loss_decomp

- The three terms of the expansion are named A, B, C and split into separate sums:

```
    U.sum (fun i =>
      (d ^ 2) * (x i) ^ 2
      + (a0 * x i - y i) ^ 2
      + (2 * d) * (x i * (a0 * x i - y i))) := by
  refine Finset.sum_congr rfl ?_
  intro i hi
  have ht := hterm i
  simp [ht, pow_two]
  ring_nf
let A : Fin n → ℝ := fun i => (d ^ 2) * (x i) ^ 2
let B : Fin n → ℝ := fun i => (a0 * x i - y i) ^ 2
let C : Fin n → ℝ := fun i => (2 * d) * (x i * (a0 * x i - y i))
```

- After splitting, $\sum A$ gives $S_{xx}(a - a^*)^2$, $\sum B$ gives $L(a^*)$, and $\sum C$ vanishes by the normal equation.

Lemma 4: aStar_minimizes

- The decomposition gives us $L(a) = L(a^*) + S_{xx}(a - a^*)^2$. Since both $S_{xx} \geq 0$ and $(a - a^*)^2 \geq 0$, the penalty is nonneg, so $L(a^*) \leq L(a)$:

```
theorem aStar_minimizes (x y : Fin n → ℝ) (h : Sxx x ≠ 0) :
    ∀ a : ℝ, loss x y (aStar x y) ≤ loss x y a := by
  classical
  intro a
  have hdecomp := loss_decomp (x := x) (y := y) h a
  have hnonneg : 0 ≤ (Sxx x) * (a - aStar x y) ^ 2 :=
    mul_nonneg (Sxx_nonneg (x := x)) (sq_nonneg (a - aStar x y))
  calc
    loss x y (aStar x y)
        ≤ loss x y (aStar x y) + (Sxx x) * (a - aStar x y) ^ 2 := by
          exact le_add_of_nonneg_right hnonneg
    _ = loss x y a := by
          exact hdecomp.symm

end LeanML.Supervised.Regression.Linear.Origin
```

Affine Case: $y = ax + b$

- With two parameters a and b, the loss becomes:

$$L(a, b) = \sum_{i=1}^{n} (ax_i + b - y_i)^2$$

- We now need additional statistics: $S_x = \sum x_i$, $S_y = \sum y_i$, and the determinant $\Delta = n \cdot S_{xx} - S_x^2$.
- The optimal parameters come from solving the normal equations:

$$\begin{pmatrix} S_{xx} & S_x \\ S_x & n \end{pmatrix} \begin{pmatrix} a \\ b \end{pmatrix} = \begin{pmatrix} S_{xy} \\ S_y \end{pmatrix}$$

- The loss decomposes as:

$$L(a, b) = L(a^*, b^*) + S_{xx}\, da^2 + 2\, S_x\, da\, db + n\, db^2$$

where $da = a - a^*$, $db = b - b^*$.

- The quadratic form equals $\sum (da \cdot x_i + db)^2$, a sum of squares, hence nonneg. This gives us $L(a^*, b^*) \leq L(a, b)$ for all a, b.

Conclusions

- We formally verified that the OLS estimators minimize the squared loss for both regression through the origin and affine regression.
- The proofs are fully machine-checked in Lean 4 with Mathlib.
- Key challenges were manipulating finite sums in Lean and bridging algebraic rewrites with tactics like `ring`, `field_simp`, and `linarith`.
- Future work includes extending to multivariate regression, ridge regression, and classification algorithms.
- Code available at: `https://github.com/itsalissonsilva/lean-ml`