

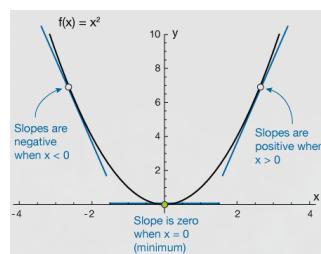
Part 1 Formula Application for Differentiation

a. Interpret dy/dx geometrically:

dy/dx represents the slope of a tangent line to a curve at a specific point.

In Linear Regression dy/dx shows how steep the loss function is at a particular point. When dy/dx is positive, the loss is trending upward or uphill. When negative the loss is decreasing or going downhill. If dy/dx is zero, we are at a flat point, potentially the minimum.

The sign tells me which direction to move the weights, either left or right on the curve. The magnitude tells me how steep the slope is, larger values result in steeper slopes.



Source: [xaktly.com – The Derivative \(Figure 9\)](http://xaktly.com – The Derivative (Figure 9))

b. How many differentiation formulas do we have and what are they?

1. Power Rule:

$$d/dx(x^n) = nx^{(n-1)}$$

(Bring down the exponent, reduce power by 1)

2. Constant:

$$d/dx[c] = 0$$

(Derivative of any constant is zero)

3. Constant Multiple:

$$d/dx[cf(x)] = (c)df/dx$$

(Pull the constant out front)

4. Sum Rule:

$$d/dx[f + g] = f'(x) + g'(x)$$

(Differentiate each term separately)

5. Product Rule:

$$d/dx[fg] = (f)dg/dx + (g)df/dx$$

(First times derivative of second, plus second times derivative of first)

6. Quotient Rule:

$$d/dx[f/g] = (g \cdot df/dx - f \cdot dg/dx) / g^2$$

(Bottom times derivative of top, minus top times derivative of bottom, all over bottom squared)

7. Chain Rule:

$$d/dx[f(g(x))] = f'(g(x)) * g'(x)$$

(Derivative of outer function times derivative of inner function)

8. Exponential:

$$d/dx[e^x] = e^x$$

(e^x is its own derivative)

c. Differentiate:

Problem 1:

$$y = 4 + 2x - 3x^2 - 5x^3 - 8x^4 + 9x^5$$

1. Apply the sum rule and differentiate each term separately

2. Constant term

$$d/dx[4] = 0$$

3. Linear term

$$d/dx[2x] = 2$$

4. Apply power rule to remaining terms

$$d/dx[-3x^2] = -6x$$

$$d/dx[-5x^3] = -15x^2$$

$$d/dx[-8x^4] = -32x^3$$

$$d/dx[9x^5] = 45x^4$$

5. Combine all terms

$$\text{Answer: } dy/dx = 2 - 6x - 15x^2 - 32x^3 + 45x^4$$

Problem 2:

$$y = 1/x + 3/x^2 + 2x^3 = x^{-1} + 3x^{-2} + 2x^{-3}$$

1. Rewrite using negative exponents

This allows us to use the power rule

2. Apply the power rule

$$d/dx[x^{-n}] = nx^{-n-1}$$

$$d/dx[x^{-1}] = -1 * x^{-2} = -x^{-2}$$

$$d/dx[3x^{-2}] = 3(-2)x^{-3} = -6x^{-3}$$

$$d/dx[2x^{-3}] = 2(-3)x^{-4} = -6x^{-4}$$

3. Combine terms

$$dy/dx = -x^{-2} - 6x^{-3} - 6x^{-4}$$

4. Convert back to fractions (Optional but it looks nice)

$$\text{Answer: } dy/dx = -1/x^2 - 6/x^3 - 6/x^4$$

Problem 3:

$$y = \sqrt[3]{(3x^2)-1}/5x$$

1. Rewrite with exponents:

$$y = (3x^2)^{(1/3)} - 1/(5x)^{(1/2)}$$

$$y = 3^{(1/3)} \cdot x^{(2/3)} - 1/(5^{(1/2)} \cdot x^{(1/2)})$$

$$y = 3^{(1/3)} \cdot x^{(2/3)} - 5^{-(-1/2)} \cdot x^{-(-1/2)}$$

2. Simplify:

$$y = 3^{(1/3)} \cdot x^{(2/3)} - (1/\sqrt{5}) \cdot x^{(-1/2)}$$

3. Apply power rule:

$$\frac{dy}{dx} = 3^{(1/3)} \cdot (2/3)x^{(-1/3)} - (1/\sqrt{5}) \cdot (-1/2)x^{(-3/2)}$$

$$\frac{dy}{dx} = (2/3) \cdot 3^{(1/3)} \cdot x^{(-1/3)} + (1/2\sqrt{5}) \cdot x^{(-3/2)}$$

4. Combine and simplify:

$$\text{Answer: } \frac{dy}{dx} = (2 \cdot \sqrt[3]{3})/(3x^{(1/3)}) + 1/(2\sqrt{5} \cdot x^{(3/2)})$$

d. Define partial derivative

A partial derivative measures the rate of change of a multivariable function with respect to one variable while holding all other variables constant.

In linear regression, partial derivatives allow us to find how the loss function changes with respect to each individual parameter (like slope or intercept) independently, which tells us how to adjust each parameter to minimize the loss.

e. Given the following functions find $\partial z/\partial x$ and $\partial z/\partial y$ **Problem 1:**

$$z = 2x^2 - 3xy + 4y^2$$

1. Find $\partial z/\partial x$: Treat y as a constant

$$\frac{\partial z}{\partial x} = \frac{\partial}{\partial x}[2x^2 - 3xy + 4y^2]$$

$$\frac{\partial z}{\partial x} = 4x - 3y + 0$$

$$\frac{\partial z}{\partial x} = 4x - 3y$$

2. Find $\partial z/\partial y$: Treat x as a constant

$$\frac{\partial z}{\partial y} = \frac{\partial}{\partial y}[2x^2 - 3xy + 4y^2]$$

$$\frac{\partial z}{\partial y} = 0 - 3x + 8y$$

$$\frac{\partial z}{\partial y} = -3x + 8y$$

$$\text{Answer: } \frac{\partial z}{\partial x} = 4x - 3y \text{ and } \frac{\partial z}{\partial y} = -3x + 8y$$

Problem 2:

$$z = x^2/y + y^2/x$$

1. Rewrite using negative exponents

$$z = x^2y^{-1} + y^2x^{-1}$$

This allows us to use the power rule

2. Find $\partial z/\partial x$: Treat y as a constant

$$\frac{\partial z}{\partial x} = 2xy^{-1} + y^2(-1)x^{-2}$$

$$\frac{\partial z}{\partial x} = 2x/y - y^2/x^2$$

3. Find $\partial z/\partial y$: Treat x as a constant

$$\frac{\partial z}{\partial y} = x^2(-1)y^{-2} + 2yx^{-1}$$

$$\frac{\partial z}{\partial y} = -x^2/y^2 + 2y/x$$

$$\text{Answer: } \frac{\partial z}{\partial x} = 2x/y - y^2/x^2 \text{ and } \frac{\partial z}{\partial y} = -x^2/y^2 + 2y/x$$

Problem 3:

$$z = e^{(x^2 + xy)}$$

1. Use chain rule for $\partial z/\partial x$

Let $u = x^2 + xy$, then $z = e^u$

$$\frac{\partial z}{\partial x} = e^u \cdot \frac{\partial u}{\partial x}$$

$$\frac{\partial u}{\partial x} = 2x + y$$

$$\frac{\partial z}{\partial x} = e^{(x^2+xy)} \cdot (2x + y)$$

2. Use chain rule for $\partial z/\partial y$

Let $u = x^2 + xy$, then $z = e^u$

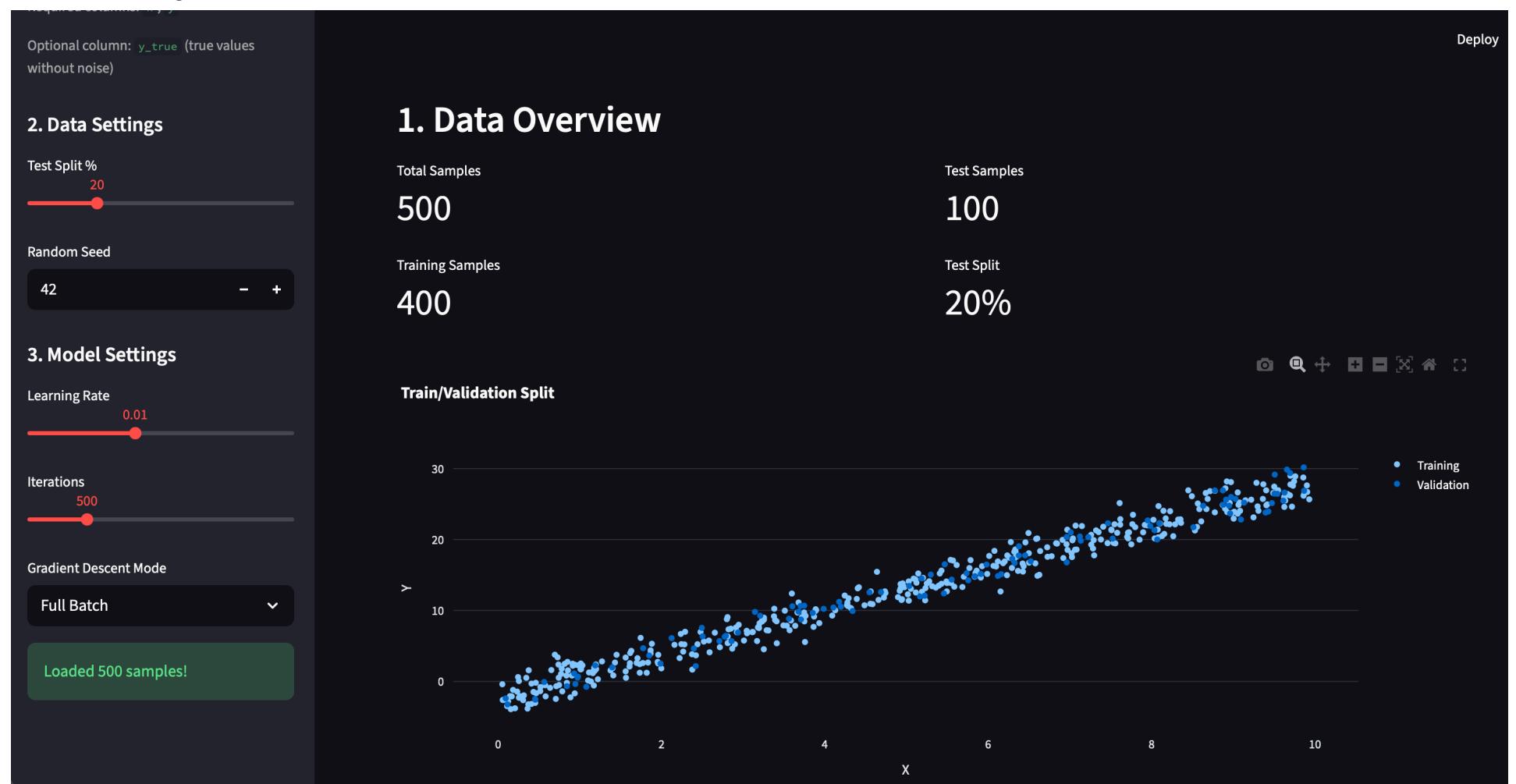
$$\frac{\partial z}{\partial y} = e^u \cdot \frac{\partial u}{\partial y}$$

$$\frac{\partial u}{\partial y} = x$$

$$\frac{\partial z}{\partial y} = e^{(x^2+xy)} \cdot x$$

$$\text{Answer: } \frac{\partial z}{\partial x} = (2x + y)e^{(x^2+xy)} \text{ and } \frac{\partial z}{\partial y} = xe^{(x^2+xy)}$$

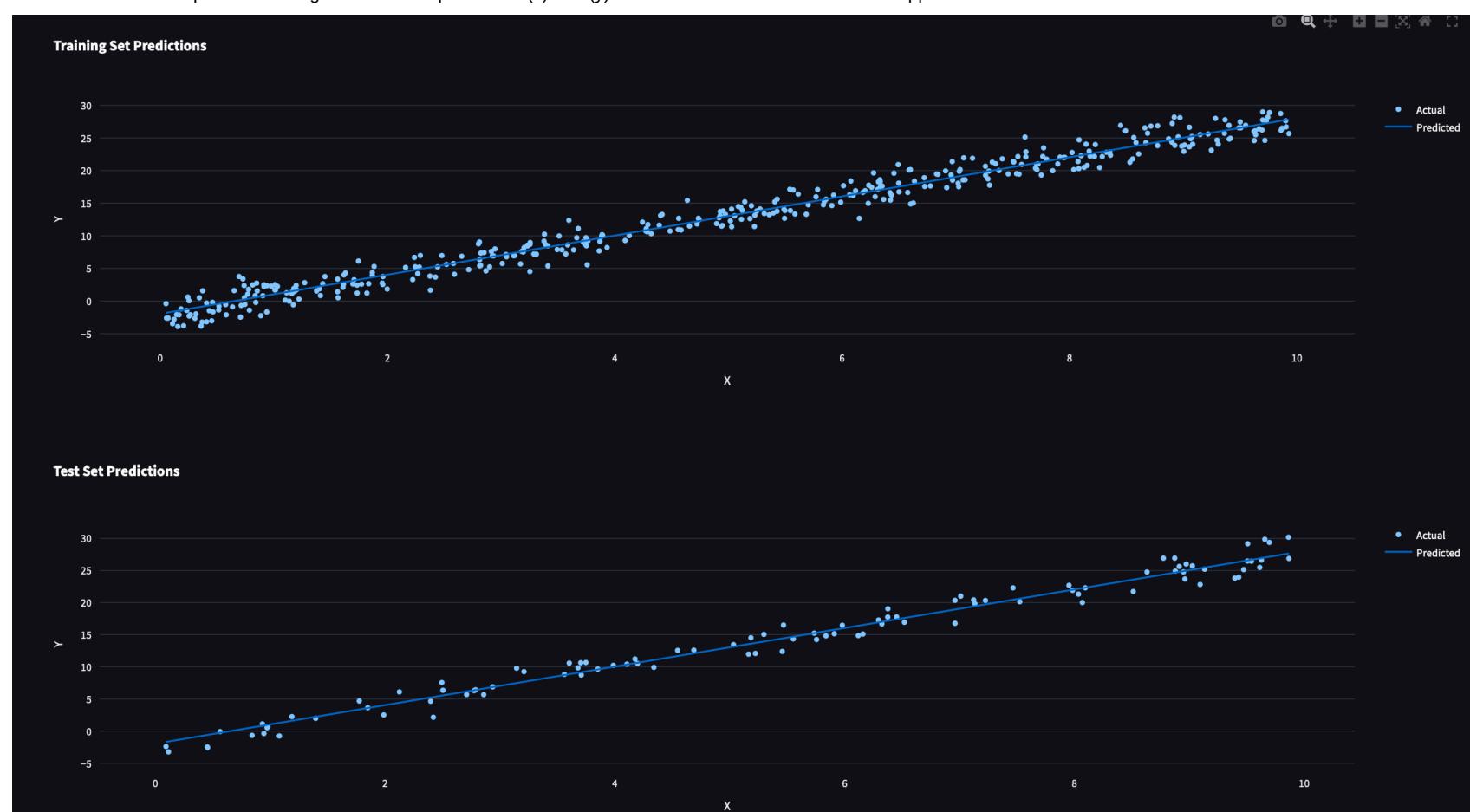
Part 2 Linear Regression



Here you can see we have created a synthetic dataset that fits the linear regression model $y = b + wx + \epsilon$.

We have included a controlled amount of noise (ϵ) to simulate real-world data scenarios.

The dataset has been plotted showing the relationship between (x) and (y) with noise included in our streamlit app.

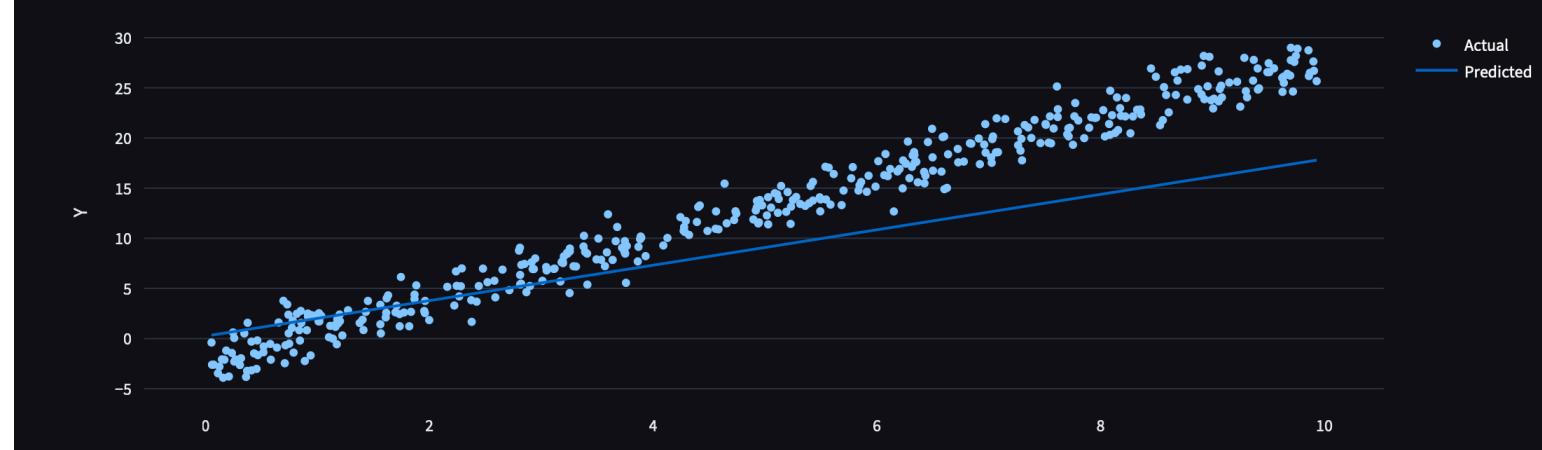


Here you can see that we have randomly partitioned our data into a train and test set.

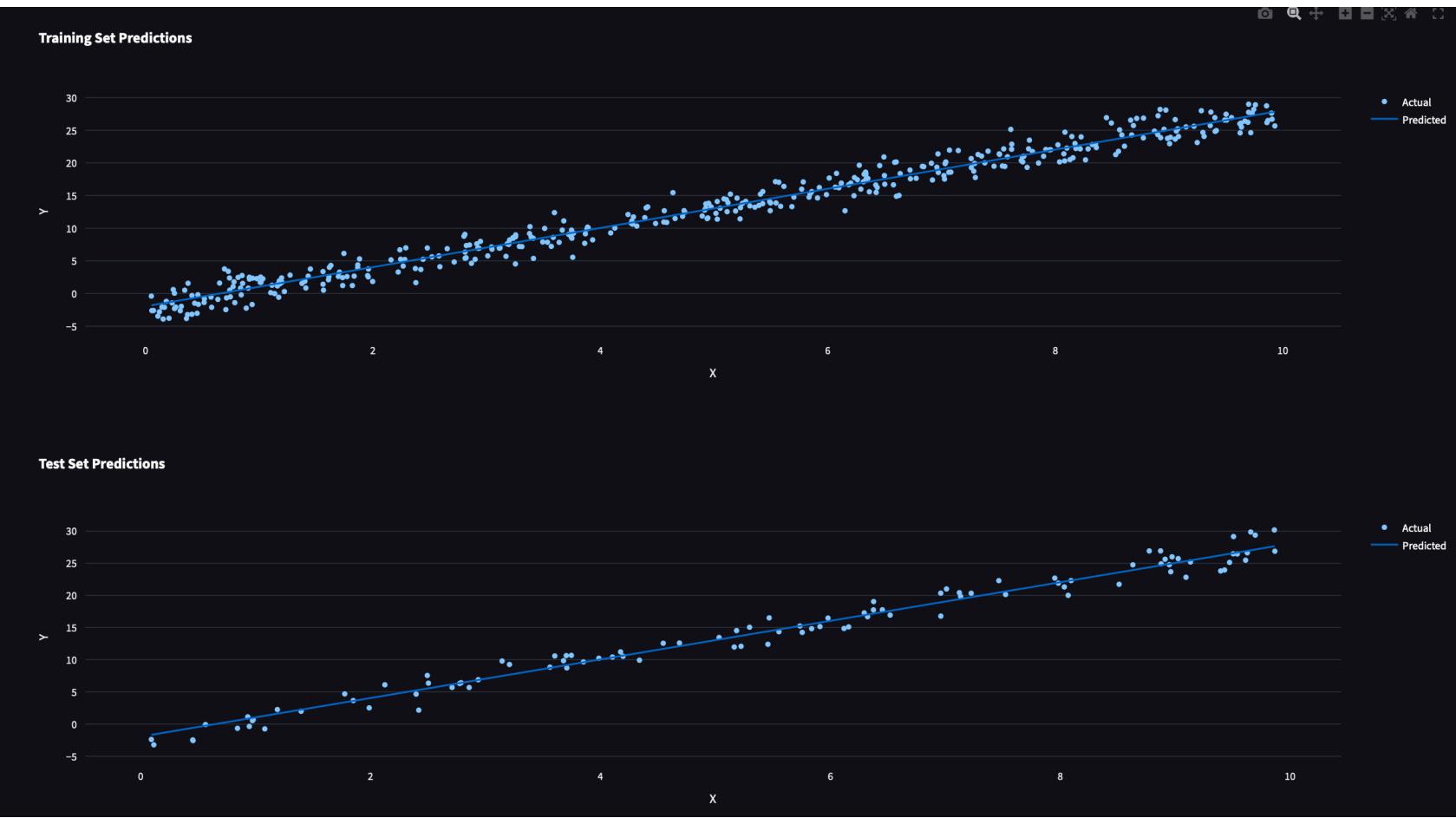
The randomness helps us avoid bias in the data.

2. Initial Model (Before Training)

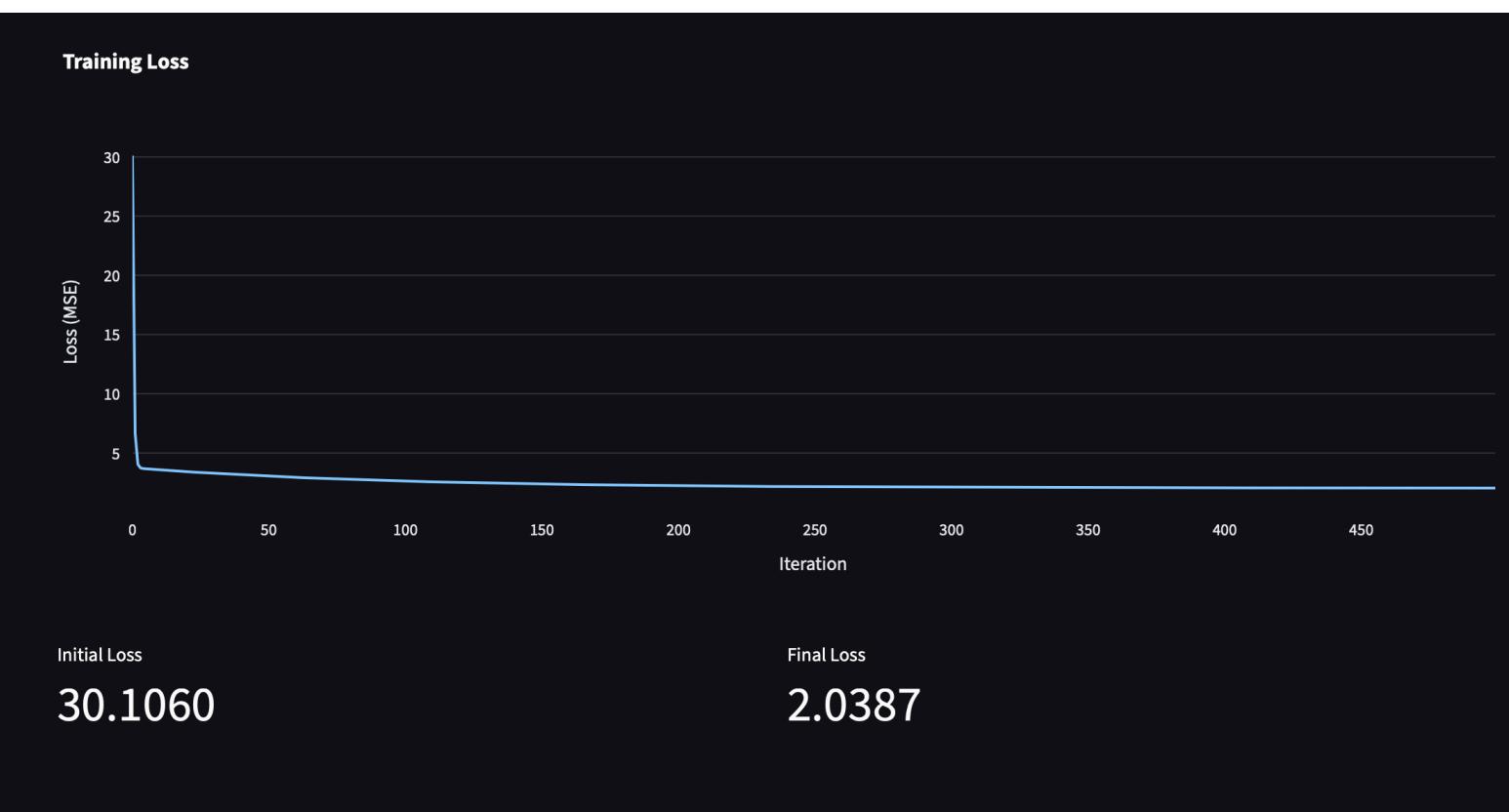
Initial Random Guess



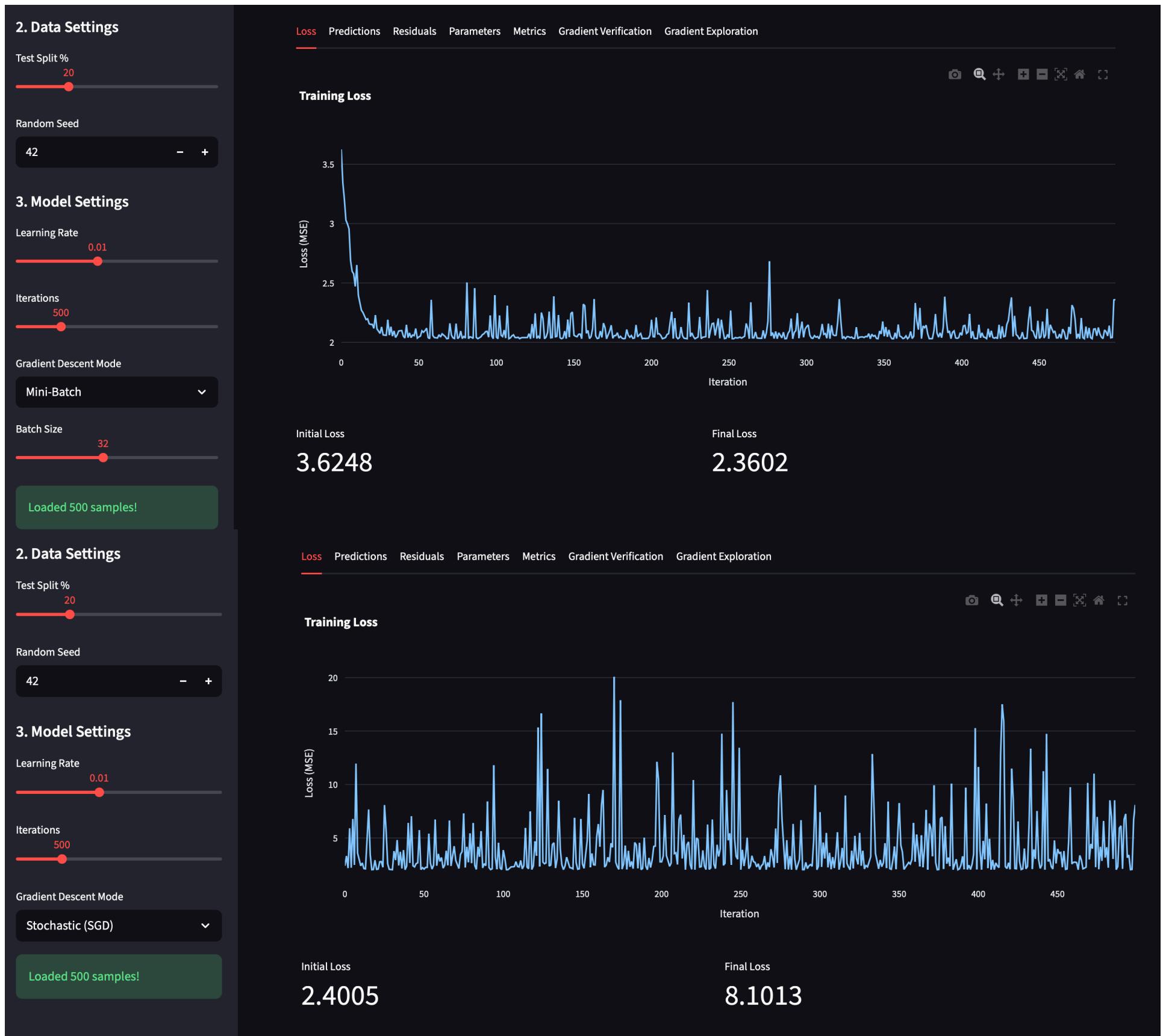
The weight and bias of the linear regression model has been initialized using random values. The initial random guess for these values should be easily reproducible since the random function has been seeded with the number 42.



This is the same image from the train test split, however here we are emphasizing the predictions made by the model on their respective splits. The model appears to generalize its learnings well on the validation set.



This shows the Mean Squared Error for our linear regression model as it trains on the synthetic dataset over time. We can see that the MSE falls very rapidly, this is primarily due to the fact that our data is highly linear and does not have much noise. This was trained using a Full Batch method of Gradient Descent.



Here we can see the linear regression model using different methods of gradient descent. The first one is mini batch and the second image is stochastic. We can see that mini batch is a bit noisier than the full batch, but still results in a lower MSE than our starting.

However, stochastic is so noisy that we actually result in a higher MSE than we started with. Immediately what this tells me is that the learning rate for the stochastic descent is too high for it to settle on a global/local minima.

3. Training Progress

Training complete!

Loss Predictions Residuals Parameters Metrics **Gradient Verification** Gradient Exploration

Numerical Gradient Verification

Comparing analytical gradients with numerical approximations (finite differences)

Weight Gradient (dw)

Analytical: -0.01393659

Numerical: -0.01393659

Difference: 1.64e-09

Bias Gradient (db)

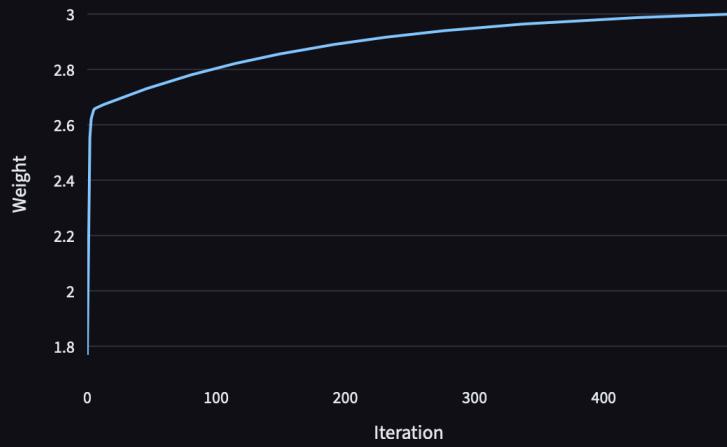
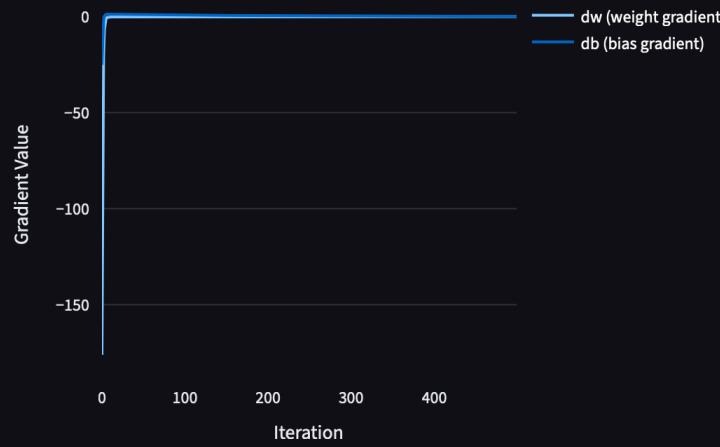
Analytical: 0.09257254

Numerical: 0.09257253

Difference: 1.21e-09

✓ Gradients verified! Analytical and numerical gradients match.

Here you can see that we have calculated the weights and biases using both the numerical method (gradient descent) as well as the analytical method (Gauss Markov Theorem). The numerical method matches very closely to what the analytical method says.

Weight Over Iterations**Gradients Over Iterations**

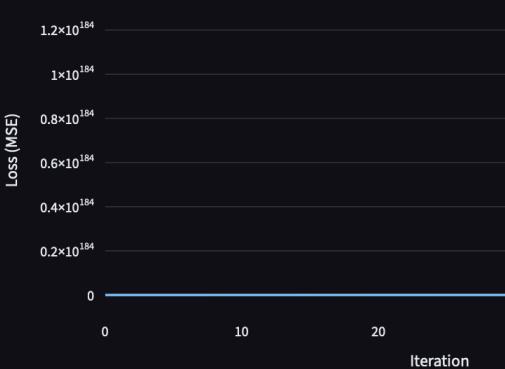
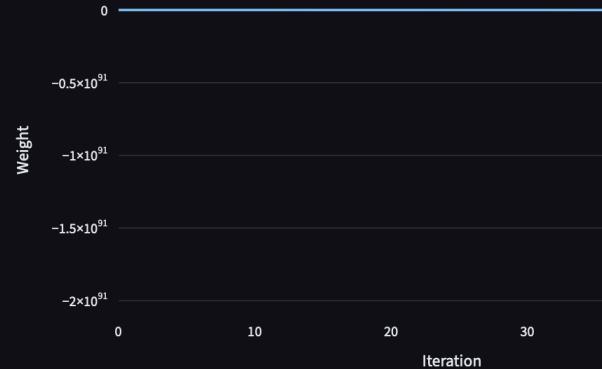
Final Weight: 2.9997

Final Bias: -1.9518

This is the gradient over time as we train the model. You can see that the gradient tunes towards the correct value (the analytical one) very quickly because the synthetic dataset we used to train on is very simple. Even though we had 500 iterations the model finds the correct parameters in the first handful of iterations.

Gradient Explosion Demonstration

What happens when the learning rate is too large?

**Training Loss****Weight Over Iterations****Observations:**

- With a large learning rate, gradients cause parameters to overshoot
- Loss may oscillate wildly or diverge to infinity (NaN)
- Weights can explode to very large values
- This is why learning rate tuning is critical!

Here is what happens when we increase the learning rate to an excessive amount. You'll notice in the graphs that we get a giant leap in our MSE value, as well as wild swings in our weight values for the model. Large learning values allow the model to move more quickly towards the correct solution, however it can also cause it to overshoot the ideal set of parameters because we don't have enough resolution in the step size to make the small adjustments the parameters need.

Ethical Considerations

Potential for bias in data generation affecting model fairness:

The bias exhibited by the model will be the same biases that we have when creating the dataset. For example if we were building an ML model to determine insurance rates based on the likelihood of a break-in our model could be subject to bias if we do not choose a random sample of housing and break-ins. If we picked only houses in Scottsdale for our dataset, our break-in prediction could be artificially low.

Ensuring data privacy when using real-world data:

when handling user data to train our models we need to be extremely careful on how we handle it. One important consideration is scrubbing any Personally Identifiable Information (PII) from the data to protect user anonymity. If we fail to do this then confidential user information could leak out of the model, especially if our model is available for public use (ex: ChatGPT)

Importance of model transparency and explainability:

Responsible use of models to benefit society and minimize harm:

Just like any piece of technology, machine learning models can be used to help people, or it can be used to hurt people. When putting a machine learning model into use we should always keep the human impact at the forefront of our mind. For example, using a machine learning model to screen job application candidates without any kind of human oversight can result in a more depersonalized job interview process, souring the job applicant. It could also have biases that would go unchecked without human oversight, like only responding to candidates with an American sounding name.

Mitigating negative impacts of prediction errors:

The primary mitigator of negative impacts of prediction is to get a dataset that most accurately represents the parent distribution. The second best way to mitigate is to keep good metrics on how the model is performing in production, and actively monitoring it to make sure that the model is performing as intended.

Program flowchart (and code to generate flowchart using mermaid.js)

