Title: Exploring Polynomial Functions Through Iterative Fitting

Objective:

This exercise is designed to help participants develop an intuitive and mathematical understanding of polynomial functions and their relationships to data points. By iteratively fitting polynomials of increasing degree to 2, 3, up to 10 randomly generated points, participants will gain insight into the geometry and algebra of polynomials, understand overfitting and underfitting, and observe how polynomial coefficients evolve as new points are added.

Key Learning Goals:

1. **Understanding Polynomial Representation:**
   - Develop familiarity with the general form of a polynomial $P(x) = a_0 + a_1 x + a_2 x^2 + \ldots + a_n x^n$.
   - Interpret the significance of polynomial coefficients and their role in shaping the curve.
2. **Incremental Data Integration:**
   - Learn how polynomials adjust to new data points as the degree increases.
   - Observe the constraints imposed by additional points and how this affects the coefficients.
3. **Geometry of Polynomials:**
   - Visualize how polynomial curves evolve to pass through given points in the x, y plane.
   - Understand how the number of points corresponds to the polynomial degree ($n + 1$ points define a unique polynomial of degree n).
4. **Behavior of Higher-Degree Polynomials:**
   - Study the complexity introduced by higher-degree polynomials and their sensitivity to small changes in points.
   - Identify overfitting and extreme oscillations with higher degrees.
5. **Numerical Computation and Curve Fitting:**
   - Practice solving systems of linear equations to determine polynomial coefficients.
   - Utilize computational tools (e.g., NumPy or sympy) to fit and extract coefficients.
6. **Critical Thinking on Polynomial Applications:**
   - Discuss real-world scenarios where polynomial fitting is useful (e.g., interpolation, approximation).
   - Explore the limitations and challenges of using high-degree polynomials.

Example Workflow:

1. Start with 2 points (e.g., $(1, 2), (3, 5)$) and fit a linear polynomial ($P(x) = ax + b$). Record and interpret the coefficients.
2. Add a third point, fit a quadratic polynomial ($P(x) = ax^2 + bx + c$), and observe changes in the curve and coefficients.
3. Repeat this process up to 10 points, fitting polynomials of increasing degree iteratively.
4. Plot the polynomial and points at each step to visualize the curve's evolution.
5. Discuss findings and patterns in the coefficients and shapes.

Key Takeaways:

By completing this exercise, participants will:

- Build a robust understanding of polynomial fitting through practical experimentation.
- Recognize the balance between model complexity and data constraints.
- Enhance their intuition about mathematical modeling and interpolation techniques.
- Strengthen their skills in data-driven problem-solving using polynomials.

Understanding Overfitting Through Iterative Polynomial Fitting

Overfitting occurs when a model becomes overly complex, capturing noise or random variations in the data rather than the underlying trend. In the context of the iterative polynomial fitting exercise, overfitting can be understood as follows:

1. Connection to Polynomial Degree and Data Points

- **Low-degree polynomials (Underfitting):**
  When we fit a polynomial with too low a degree for the number of points (e.g., using a linear polynomial for more than two points), the curve will fail to pass through all the points. This is an example of underfitting, where the model cannot capture the complexity of the data.
- **High-degree polynomials (Overfitting):**
  As we fit higher-degree polynomials, especially when the degree approaches the number of

points minus one ($n-1$ for n points), the polynomial will pass through every point exactly. While this ensures a perfect fit to the training points, it often results in erratic and oscillatory behavior between the points, failing to generalize to unseen data.

2. Observing Overfitting in the Iterative Process

Example:

1. **Two Points:**
   Fitting a linear polynomial to two points gives a straight line. The model is simple, and no overfitting occurs because the degree matches the data's complexity.
2. **Three Points:**
   Fitting a quadratic polynomial to three points creates a smooth parabola that passes through all points without introducing unnecessary complexity.
3. **Ten Points:**
   Fitting a 9th-degree polynomial to 10 points may result in:
   - Perfectly passing through all the points.
   - Highly oscillatory behavior between the points, with extreme peaks and valleys.

Visualization:

When plotted, the high-degree polynomial (9th degree for 10 points) may deviate wildly between points, highlighting overfitting. In contrast, a lower-degree polynomial (e.g., quadratic or cubic) would show a smoother curve that captures the general trend but doesn't necessarily pass through every point.

3. Overfitting Characteristics

- **High Sensitivity to Small Changes in Data:**
  A small perturbation in any data point leads to significant changes in the high-degree polynomial, reflecting the model's lack of robustness.
- **Complex Coefficients:**
  The coefficients of high-degree polynomials are often large and unstable, contributing to erratic behavior.
- **Poor Generalization:**
  While the high-degree polynomial fits the training points perfectly, it fails to predict unseen data

accurately due to its over-reliance on exact training points.

4. Mitigating Overfitting

In the iterative polynomial fitting exercise:

- **Early Stopping:**
  Stop increasing the polynomial degree when it begins to exhibit erratic behavior. For example, if the general trend is captured well by a cubic or quartic polynomial, avoid moving to higher degrees.
- **Regularization:**
  Introduce penalties for high-degree coefficients to discourage overfitting, ensuring a smoother curve.
- **Cross-validation:**
  Use a subset of points to validate the polynomial's generalization. If the error on validation points increases with a higher degree, it's a sign of overfitting.

5. Insights from the Exercise

This example provides a hands-on way to see:

- The transition from underfitting to overfitting.
- How model complexity relates to data constraints.
- The importance of balancing fitting accuracy with generalizability.

Overfitting is vividly observed as we increase polynomial degree, making it an excellent conceptual and practical tool for understanding this fundamental machine learning challenge.