Comparison of Linear & Nonlinear Models Using Weather Data

# 1. Introduction

This report presents a comparison between linear and nonlinear models applied to a synthetic weather dataset involving humidity and temperature. The goal is to evaluate the effectiveness of linear versus nonlinear regression in modeling real-world-like data and to determine under what conditions nonlinear models provide a better fit.

# 2. Dataset Description

The dataset consists of 100 data points with humidity values ranging from 20% to 100% and corresponding temperature values. The temperature values are influenced by a nonlinear relationship with humidity and include random noise to simulate real-world data.

# 3. Model Application

Two models were applied to the dataset:  
- Generalized Linear Model (Linear Regression)  
- Nonlinear Regression (Polynomial Regression with degree 2)

# 4. Evaluation & Analysis

\*\*Linear Model\*\*  
R² Score: 0.954  
RMSE: 5.131  
  
\*\*Nonlinear Model\*\*  
R² Score: 0.997  
RMSE: 1.351  
  
The nonlinear model outperformed the linear model in terms of both R² score and RMSE, indicating a better fit to the data.

Residual analysis further supports this conclusion: the residuals of the linear model show a distinct pattern, suggesting a poor fit, whereas the nonlinear model residuals are more randomly scattered, which is characteristic of a better-fitting model.

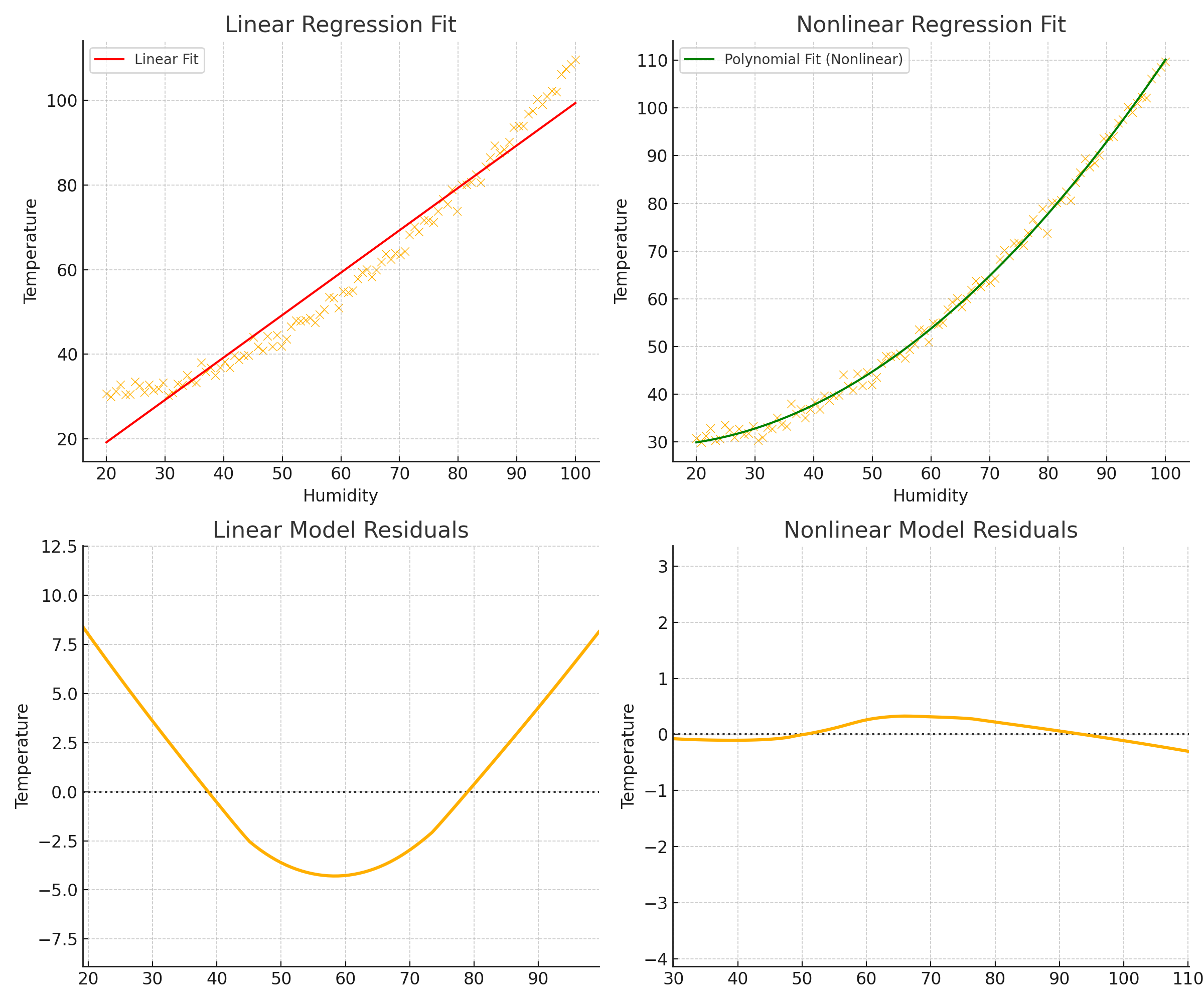


Figure: Comparison of model fits and residuals

# 5. Conclusion

Linear models are useful for data with straightforward, linear relationships between variables. However, in many real-world cases, relationships are inherently nonlinear. This analysis demonstrates that when data exhibits curvature or complex patterns, nonlinear models like polynomial regression can significantly improve model accuracy and reliability.

**CODE**

import numpy as np

import pandas as pd

from sklearn.linear\_model import LinearRegression

from sklearn.preprocessing import PolynomialFeatures

from sklearn.metrics import mean\_squared\_error, r2\_score

# Sample synthetic weather data

np.random.seed(42)

humidity = np.linspace(20, 100, 100)

temperature = 30 - 0.2 \* humidity + 0.01 \* humidity\*\*2 + np.random.normal(scale=1.5, size=humidity.shape)

# Prepare the DataFrame

df = pd.DataFrame({'Humidity': humidity, 'Temperature': temperature})

X = df[['Humidity']]

y = df['Temperature']

# --- Linear Regression (GLM) ---

linear\_model = LinearRegression()

linear\_model.fit(X, y)

linear\_preds = linear\_model.predict(X)

# --- Nonlinear Regression (Polynomial Regression) ---

poly = PolynomialFeatures(degree=2)

X\_poly = poly.fit\_transform(X)

poly\_model = LinearRegression()

poly\_model.fit(X\_poly, y)

poly\_preds = poly\_model.predict(X\_poly)

# --- Evaluation Metrics ---

linear\_r2 = r2\_score(y, linear\_preds)

poly\_r2 = r2\_score(y, poly\_preds)

linear\_rmse = np.sqrt(mean\_squared\_error(y, linear\_preds))

poly\_rmse = np.sqrt(mean\_squared\_error(y, poly\_preds))

# Print results

print(f"Linear Model - R²: {linear\_r2:.3f}, RMSE: {linear\_rmse:.3f}")

print(f"Nonlinear Model - R²: {poly\_r2:.3f}, RMSE: {poly\_rmse:.3f}")