Polynomial Regression

CS109A Introduction to Data Science

Pavlos Protopapas, Natesh Pillai

Lecture Outline

Announcements

Q&A

Part A:

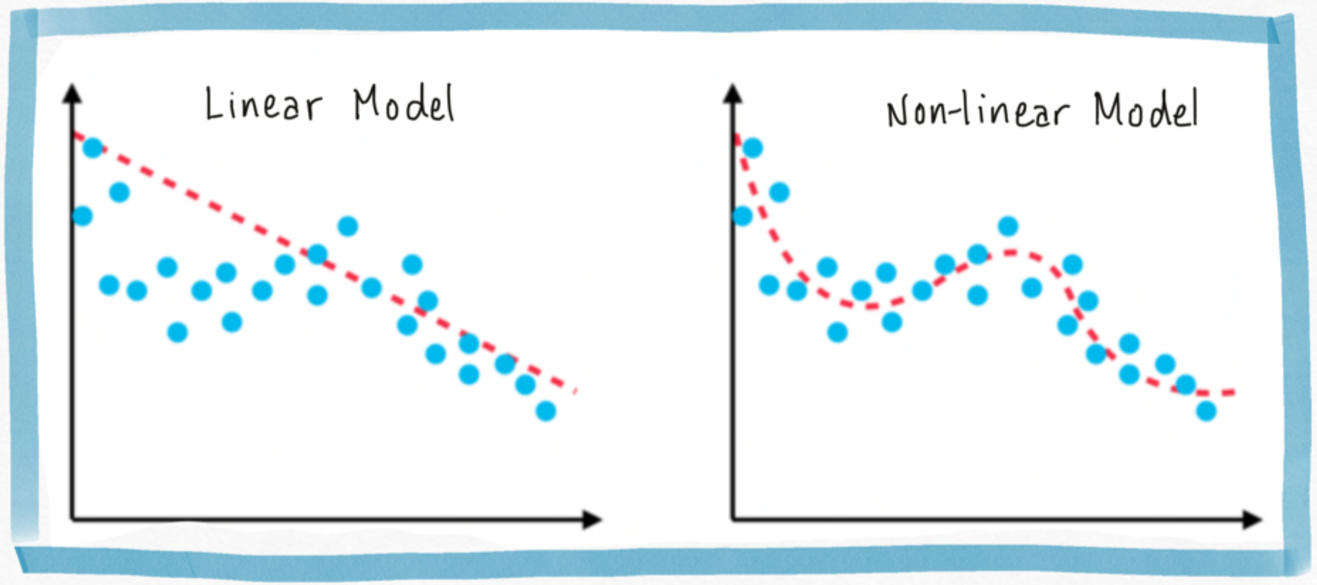
Multi-linear regression

Part B:

Polynomial Regression

Fitting non-linear data

Multi-linear models can fit large datasets with many predictors. But the relationship between predictor and target isn’t always linear.



We want a model:

$$y = f_\beta(x)$$

Where f is a non-linear function and β is a vector of the parameters of f.

Polynomial Regression

The simplest non-linear model we can consider, for a response *Y* and a predictor *X*, is a polynomial model of degree *M*,

$$y = \beta_0 + \beta_1 x + \beta_2 x^2 + \cdots + \beta_M x^M$$

Just as in the case of linear regression with cross terms, polynomial regression is a special case of linear regression - we treat each x^m as a separate predictor. Thus, we can write the *design matrix* as:

$$\mathbf{Y} = \begin{pmatrix} y_1 \\ \vdots \\ y_n \end{pmatrix}, \qquad \mathbf{X} = \begin{pmatrix} 1 & x_1^1 & \dots & x_1^M \\ 1 & x_2^1 & \dots & x_2^M \\ \vdots & \vdots & \ddots & \vdots \\ 1 & x_n & \dots & x_n^M \end{pmatrix}, \qquad \boldsymbol{\beta} = \begin{pmatrix} \beta_0 \\ \beta_1 \\ \vdots \\ \beta_M \end{pmatrix}.$$

Polynomial Regression

$$\mathbf{Y} = \begin{pmatrix} y_1 \\ \vdots \\ y_n \end{pmatrix}, \qquad \mathbf{X} = \begin{pmatrix} 1 & x_1^1 & \dots & x_1^M \\ 1 & x_2^1 & \dots & x_2^M \\ \vdots & \vdots & \ddots & \vdots \\ 1 & x_n & \dots & x_n^M \end{pmatrix}, \qquad \boldsymbol{\beta} = \begin{pmatrix} \beta_0 \\ \beta_1 \\ \vdots \\ \beta_M \end{pmatrix}.$$

This looks a lot like multi-linear regression where the predictors are powers of x!

Multi-Regression

$$\mathbf{Y} = \begin{pmatrix} y_1 \\ \vdots \\ y_y \end{pmatrix}, \qquad \mathbf{X} = \begin{pmatrix} 1 & x_{1,1} & \dots & x_{1,J} \\ 1 & x_{2,1} & \dots & x_{2,J} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & x_{n,1} & \dots & x_{n,J} \end{pmatrix}, \qquad \boldsymbol{\beta} = \begin{pmatrix} \beta_0 \\ \beta_1 \\ \vdots \\ \beta_J \end{pmatrix},$$

Model Training

We can also perform multi-polynomial regression in the same way

Give a dataset $\{(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)\}$, to find the optimal polynomial model:

$$y = \beta_0 + \beta_1 x + \beta_2 x^2 + \cdots + \beta_M x^M$$

1. We transform the data by adding new predictors:

$$\tilde{x} = [1, \tilde{x}_1, \tilde{x}_2, \dots, \tilde{x}_M]$$

where $\tilde{x}_k = x^k$

We can generate $\tilde{x}$ by calling:

`sklearn.preprocessing.PolynomialFeatures(degree=?)`

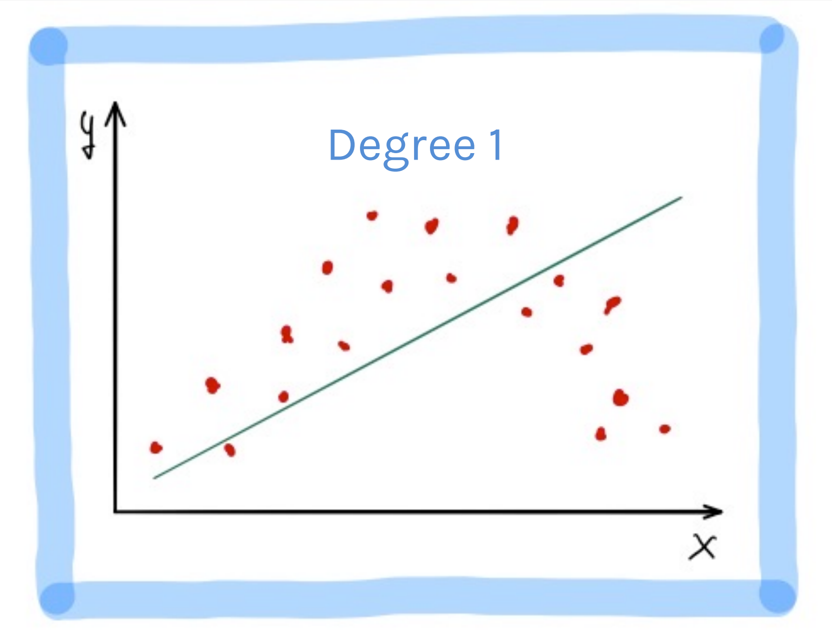
2. We find the parameter by minimizing the MSE using vector calculus yields, as in multi-linear regression

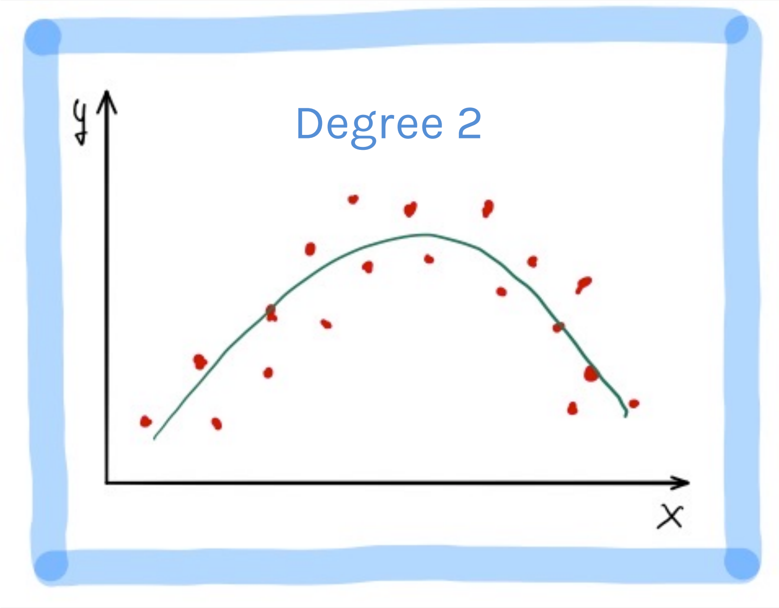
$$\boxed{\widehat{\boldsymbol{\beta}} = \left(\widetilde{X}^T \widetilde{X}\right)^{-1} \widetilde{X}^T \boldsymbol{y}}$$

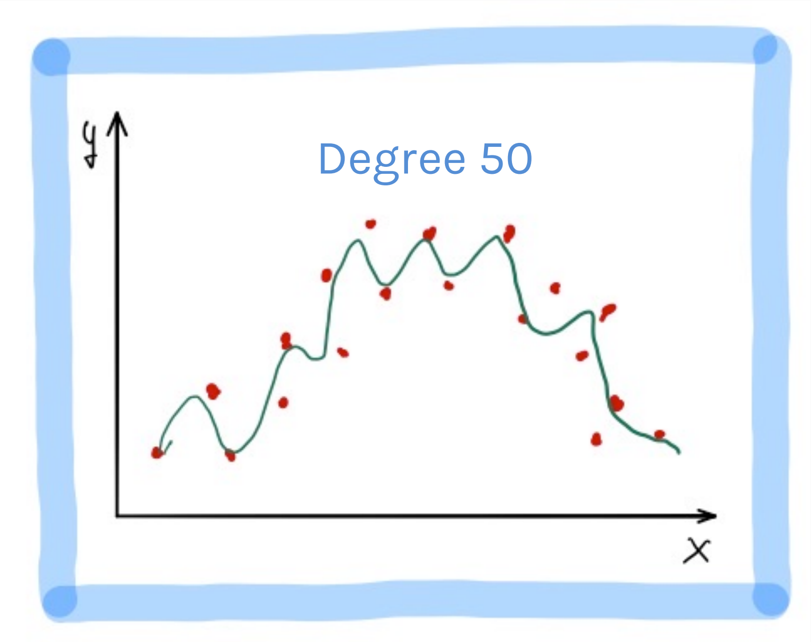
`sklearn.linear_model.LinearRegression.fit()`

Polynomial Regression (cont)

Fitting a polynomial model requires choosing a degree.







Underfitting: when the degree is too low, the model cannot fit the trend.

We want a model that fits the trend and ignores the noise.

Overfitting: when the degree is too high, the model fits all the noisy data points.

Feature Scaling

Do we need to scale out features for polynomial regression?

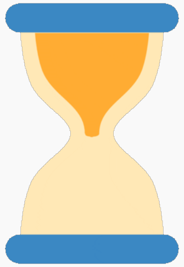
Linear regression, $Y = X\beta$, is invariant under scaling. If X is multiplied by some number λ, then β will be scaled by $\frac{1}{\lambda}$ and MSE will be identical.

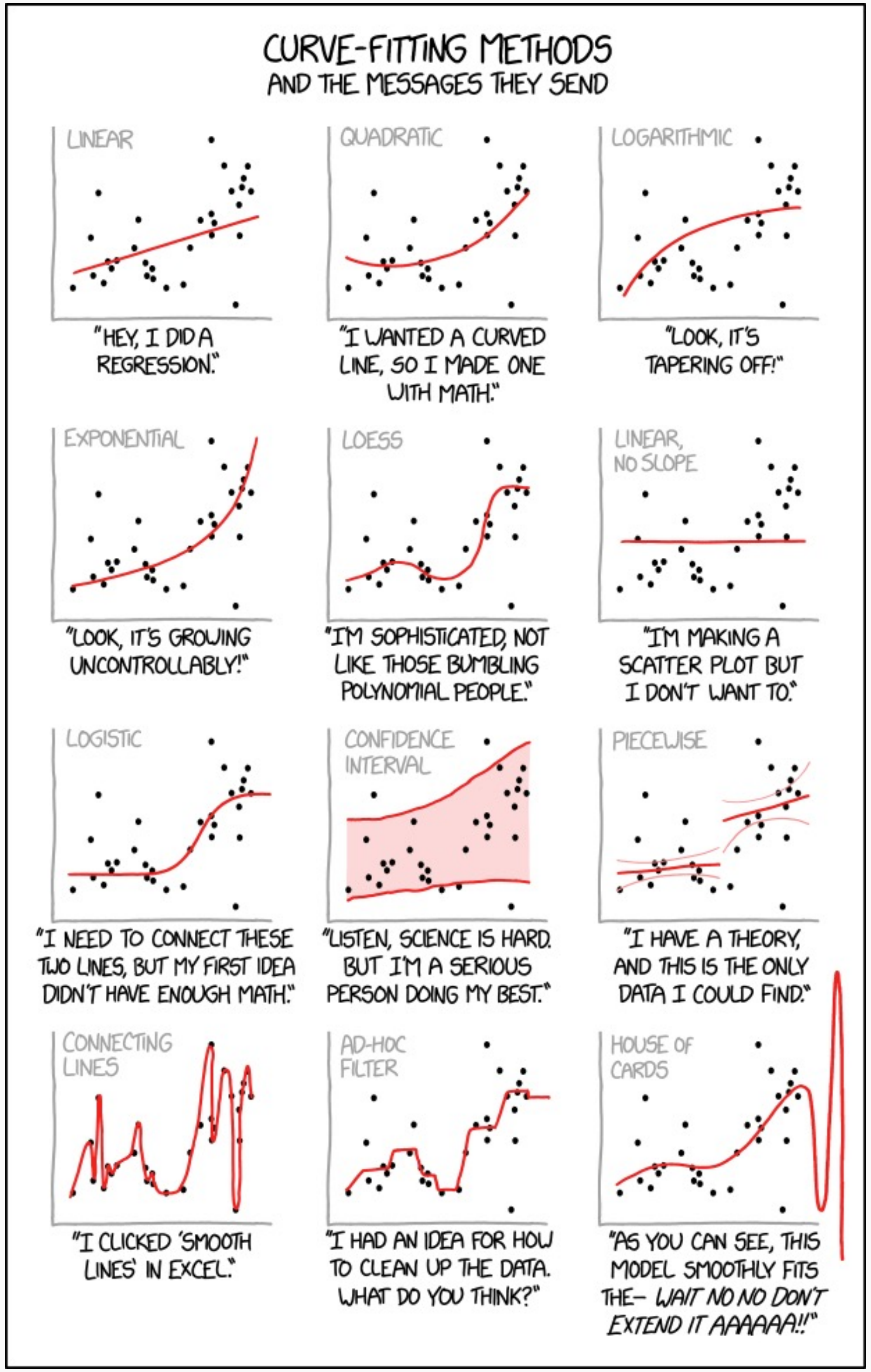
However, if the range of X is small or large, then we run into troubles. Consider a polynomial degree of 20 and the maximum or minimum value of any predictor is large or small. Those numbers to the 20$^{\text{th}}$ power will be problematic.

It is always a good idea to scale X when considering polynomial regression:

$$X^{norm} = \frac{X - \bar{X}}{\sigma_X}$$

Note: sklearn's `StandardScaler()` can do this.

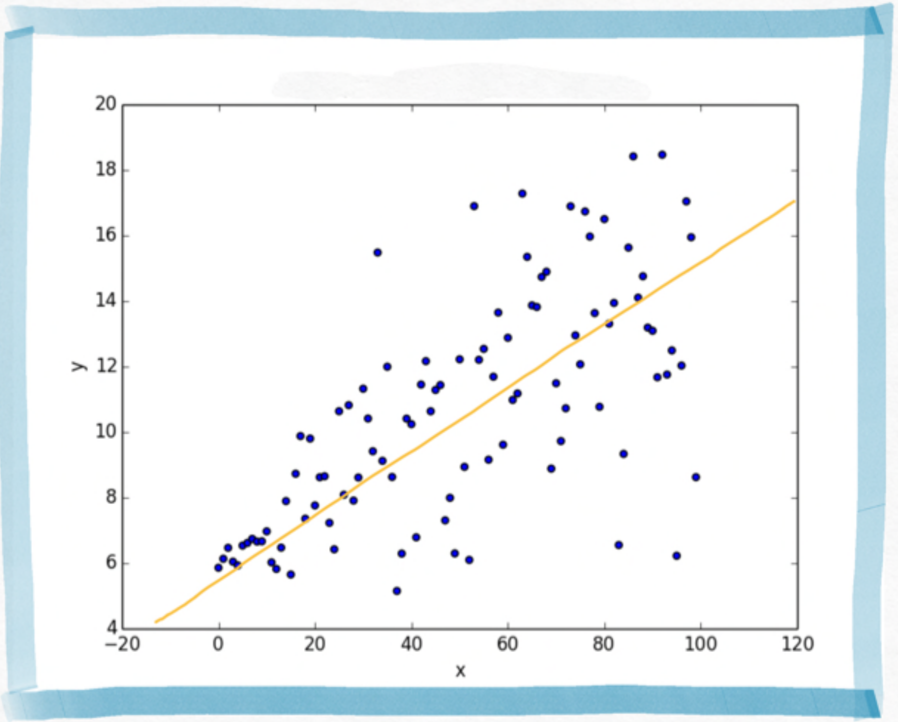


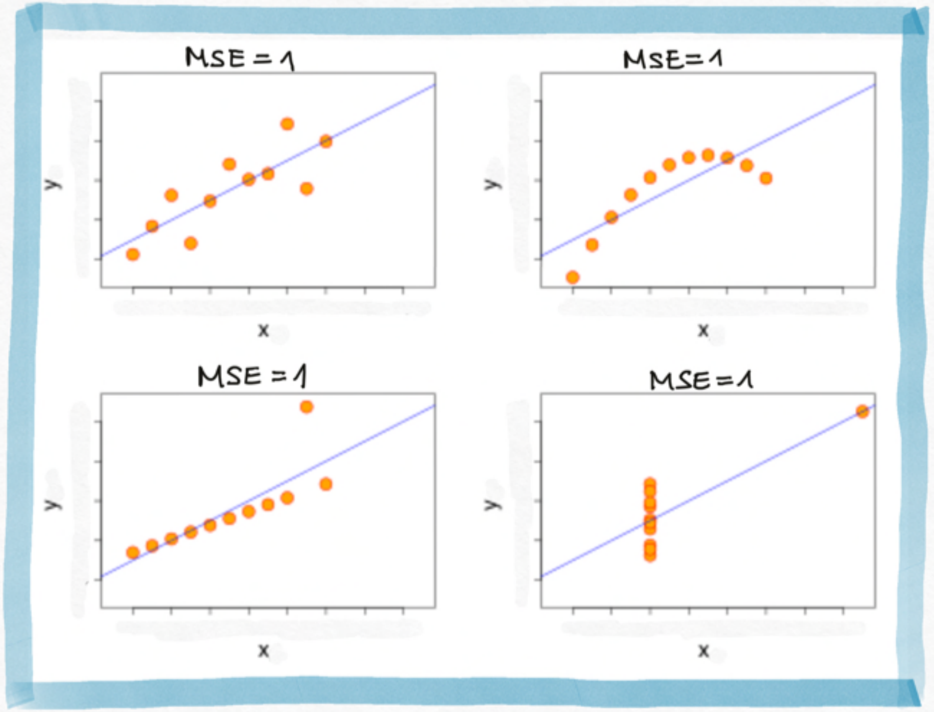

CURVE-FITTING METHODS
AND THE MESSAGES THEY SEND
LINEAR
"HEY, I DID A REGRESSION."
QUADRATIC
"I WANTED A CURVED LINE, SO I MADE ONE WITH MATH."
LOGARITHMIC
"LOOK, IT'S TAPERING OFF!"
EXPONENTIAL
"LOOK, IT'S GROWING UNCONTROLLABLY!"
LOESS
"I'M SOPHISTICATED, NOT LIKE THOSE BUMBLING POLYNOMIAL PEOPLE."
LINEAR, NO SLOPE
"I'M MAKING A SCATTER PLOT BUT I DON'T WANT TO."
LOGISTIC
"I NEED TO CONNECT THESE TWO LINES, BUT MY FIRST IDEA DIDN'T HAVE ENOUGH MATH."
CONFIDENCE INTERVAL
"LISTEN, SCIENCE IS HARD. BUT I'M A SERIOUS PERSON DOING MY BEST."
PIECEWISE
"I HAVE A THEORY, AND THIS IS THE ONLY DATA I COULD FIND."
CONNECTING LINES
"I CLICKED 'SMOOTH LINES' IN EXCEL."
AD-HOC FILTER
"I HAD AN IDEA FOR HOW TO CLEAN UP THE DATA. WHAT DO YOU THINK?"
HOUSE OF CARDS
"AS YOU CAN SEE, THIS MODEL SMOOTHLY FITS THE– WAIT NO NO DON'T EXTEND IT AAAAAA!!"

High degree of polynomial
leads to **OVERFITTING!**

Evaluation: Training Error

Just because we found the model that minimizes the squared error it doesn't mean that it's a good model. We investigate the R2 but also:



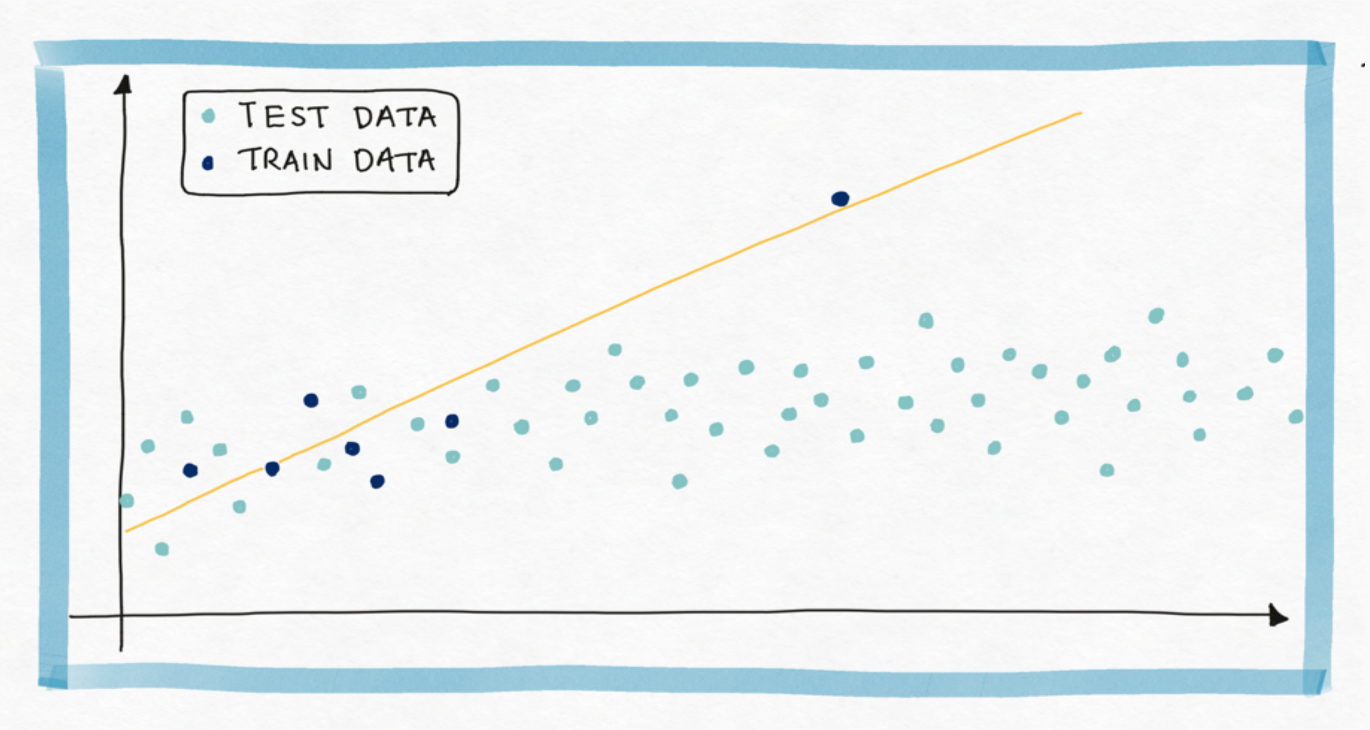


The MSE is high due to noise in the data.

The MSE is high in all four models, but the models are not equal.

Evaluation: Test Error

We need to evaluate the fitted model on new data, data that the model did not train on, the test data.



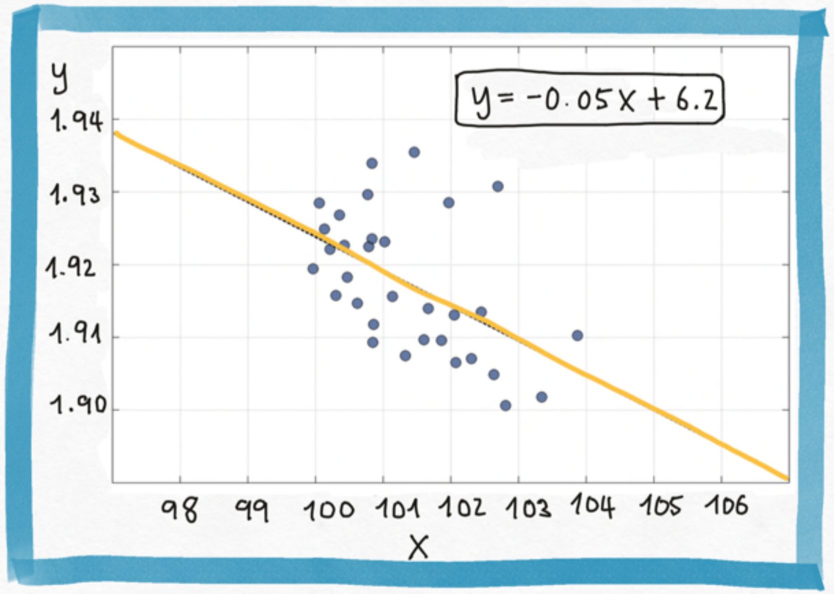
The training MSE here is 2.0 where the test MSE is 12.3.

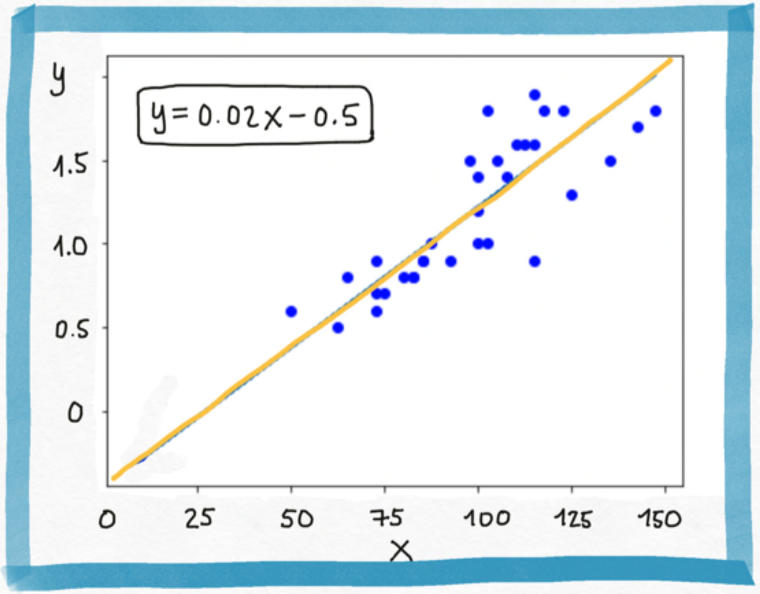
The training data contains a strange point – an outlier – which confuses the model.

Fitting to meaningless patterns in the training is called **overfitting.**

Evaluation: Model Interpretation

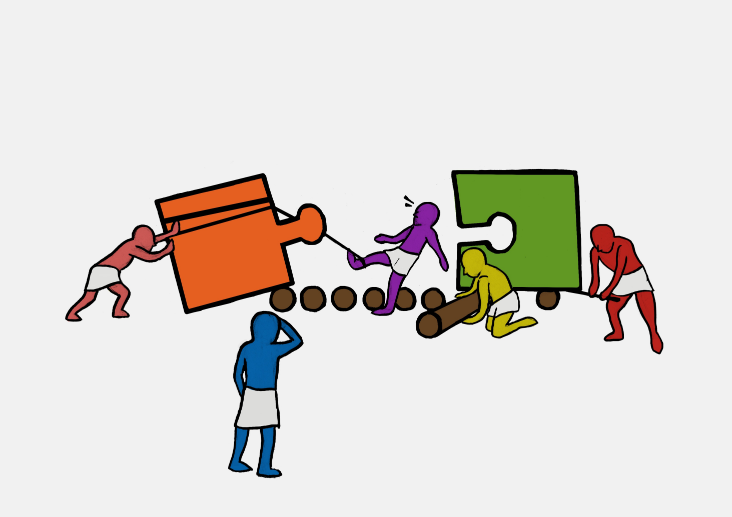
For linear models it's important to interpret the parameters





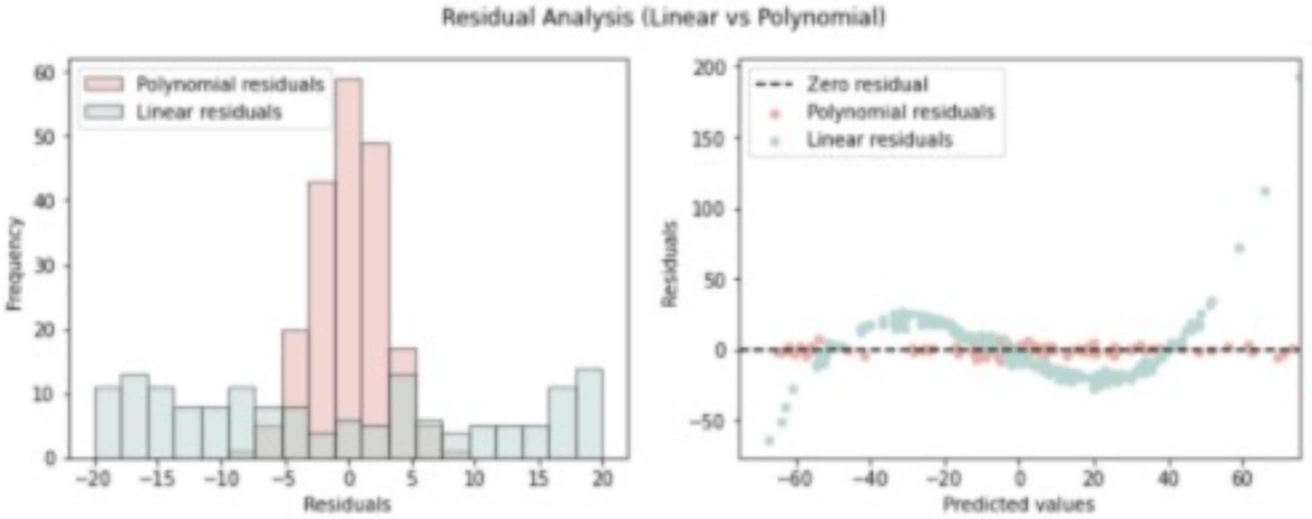
The MSE of this model is very small. But the slope is -0.05. That means the larger the budget the less the sales.

The MSE is very small, but the intercept is -0.5 which means that for very small budget we will have negative sales.

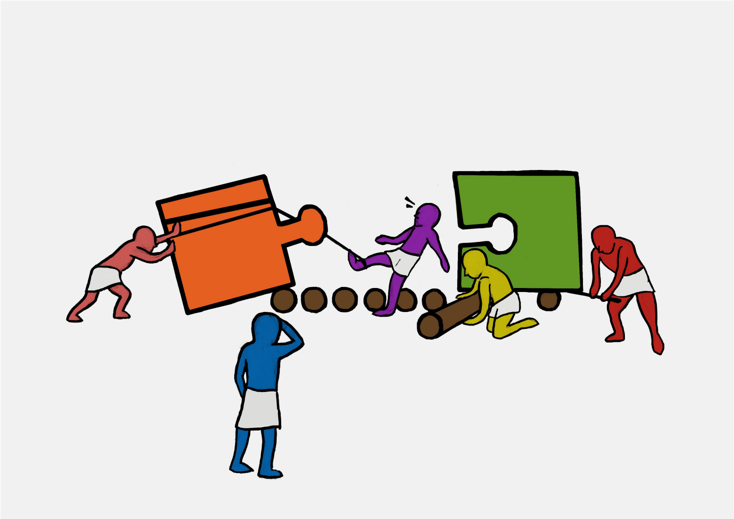


Exercise: Linear and Polynomial Regression with Residual Analysis

The goal of this exercise is to fit linear regression and polynomial regression to the given data. Plot the fit curves of both the models along with the data and observe what the residuals tell us about the two fits.



Instructions



🏋️ Exercise: Multi-collinearity vs Model Predictions

The goal of this exercise is to see how multi-collinearity can affect the predictions of a model.

For this, perform a multi-linear regression on the given dataset and compare the coefficients with those from simple linear regression of the individual predictors.