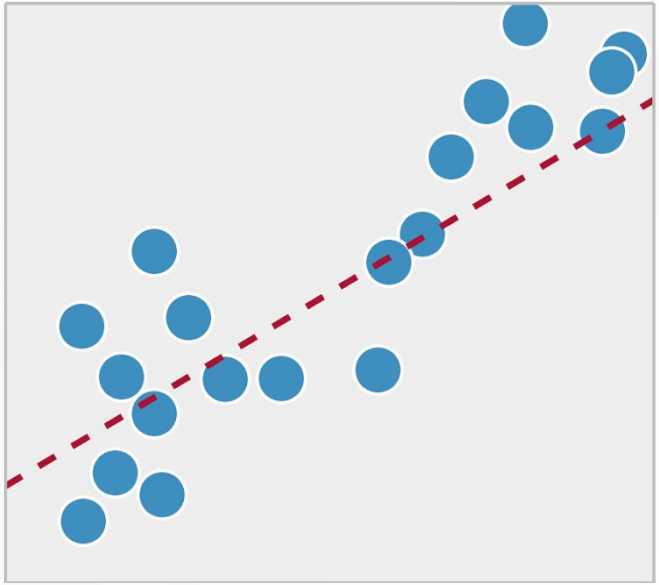
Regression Algorithms & Use-Cases

Analytiks

Regression Overview

- Form of predictive modelling technique
- Investigates the relationship between a target variable and its predictor variable(s)
- Predicts continuous values
- Used for forecasting, time series modelling and finding the causal effect relationship between the variables

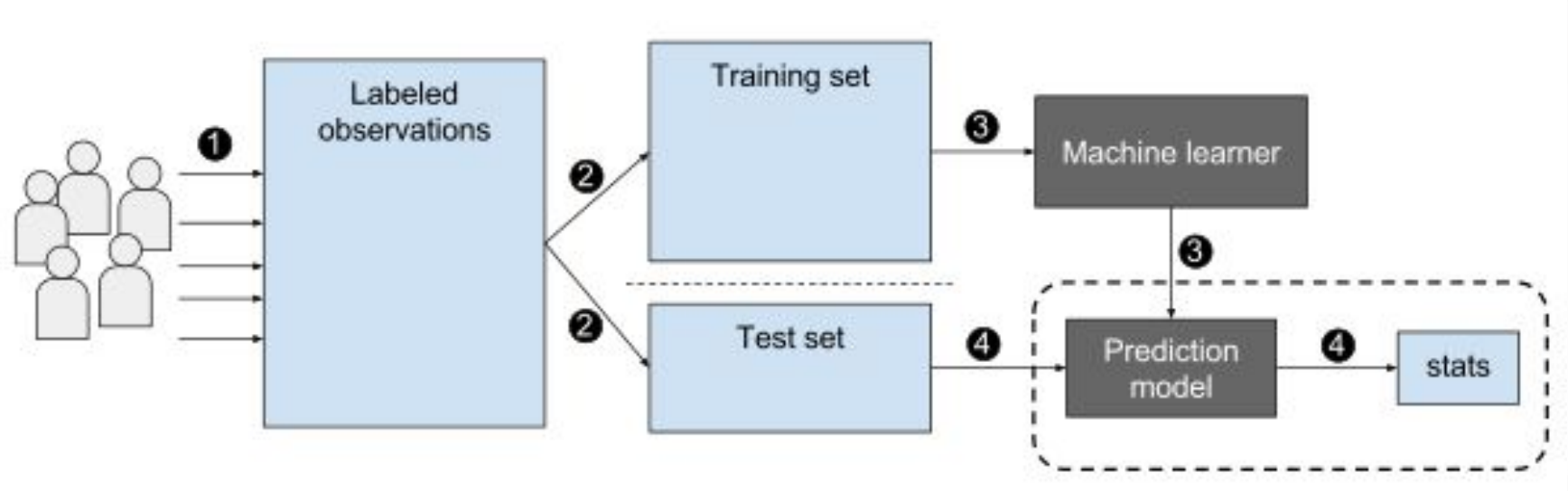


Examples:

Customer Lifetime Value	Energy Consumption	Insurance Pricing
Predicting Property Prices	Flight Time Prediction	Predicting Profitability

First, A Review

How does a machine learn?



Algorithms

Regression

Algorithms

- Linear Regression
  - Simple Linear Regression
  - Multiple Linear Regression
- Polynomial Regression
- *Logistic Regression*

Linear Regression: Overview

- Uses a linear function to predict the (average) numerical value of Y for a given value of X using a straight line (called the regression line)

$$y = \beta_0 + \beta_1 x$$

- Prediction is performed by plugging in the X values into the linear function

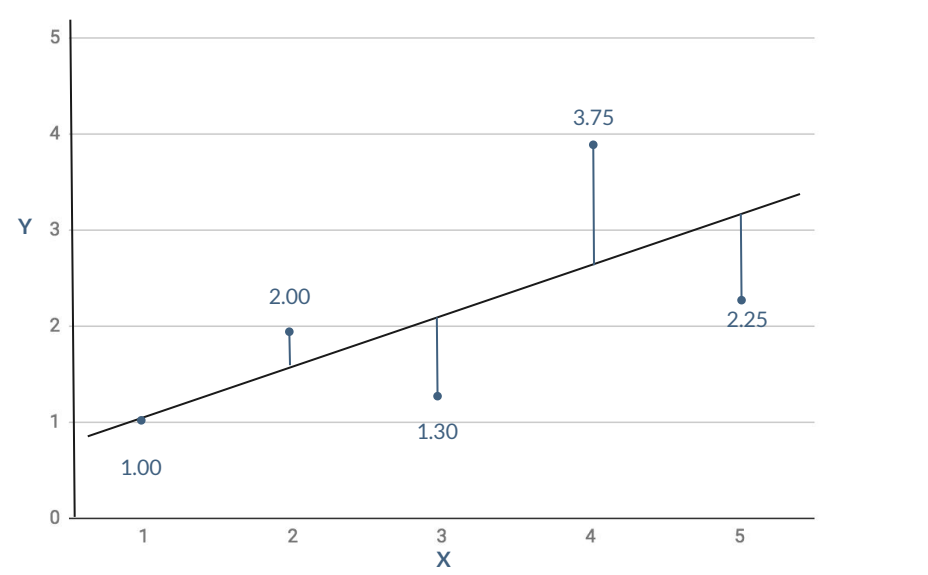
- Simple Linear Regression
    - 1 predictor variable

- Multiple Linear Regression
    - 2 or more predictor variables

Linear Regression: Regression Line

What is meant by the “Best Fitting Line”?

The line that minimizes the sum of squared errors prediction



X	Y	Y'	Y-Y'	$(Y-Y')^2$
1.00	1.00	1.210	-0.210	0.044
2.00	2.00	1.635	0.365	0.133
3.00	1.30	2.060	-0.760	0.578
4.00	3.75	2.485	1.265	1.600
5.00	2.25	2.910	-0.660	0.436

Linear Regression: Simple Linear Regression

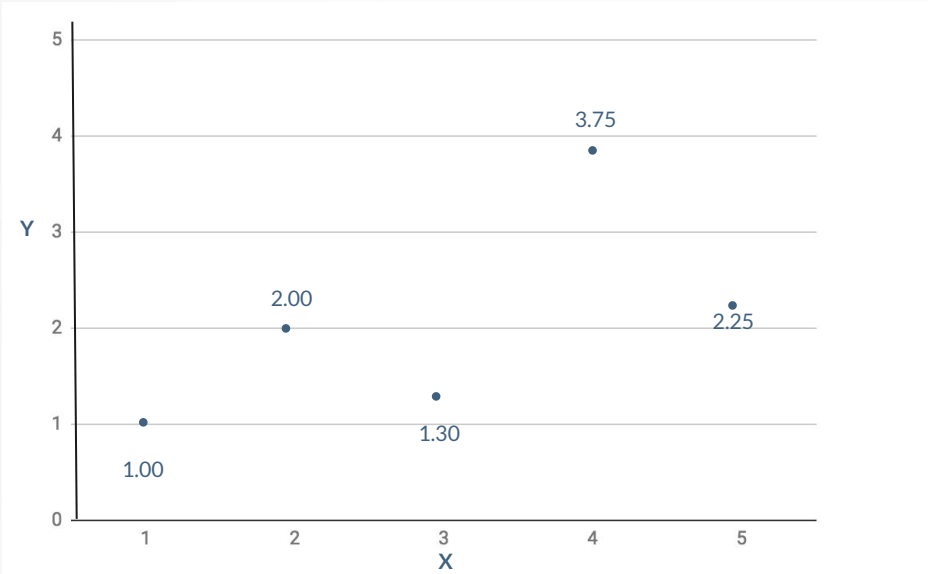
- Continuous response is modeled as a linear combination of the features:

$$y = \beta_0 + \beta_1 x$$

  - Where:
    - y - target/predicted value
    - β_0 - y-intercept
    - β_1 - slope or coefficient

- Predicts scores on the class/target variable (Y) from values of the predictor variable (X)
  - Only ONE predictor variable

X	Y
1.00	1.00
2.00	2.00
3.00	1.30
4.00	3.75
5.00	2.25



Linear Regression: Simple Linear Regression

Formula:	Example:
$y = \beta_0 + \beta_1 x$ Where: • y - target/predicted value • β_0 - y-intercept • β_1 - slope or coefficient	$y = 0.785 + 0.425x$ For x = 1: $y = 0.785 + (0.425)(1)$ $y = 1.21$ For x = 2: $y = 0.785 + (0.425)(2)$ $y = 1.64$

Linear Regression: Simple Linear Regression

Computing the regression line

Typically computed with Statistical Software

M_X	M_Y	S_X	S_Y	r
3.00	2.06	1.581	1.072	0.627

Where:

- M_X = mean of X
- M_Y = mean of Y
- S_X = standard deviation of X
- S_Y = standard deviation of Y
- r = correlation between X and Y

$$m = r\left(\frac{s_y}{s_x}\right) \qquad b = \bar{y} - m\bar{x}$$

$m = \beta_1$ $\qquad$ $b = \beta_0$

$$s_x = \sqrt{\frac{\sum_{i=1}^{n}(x_i - \bar{x})^2}{n-1}}$$

n = The number of data points

$\bar{x}$ = The mean of the x_i

x_i = Each of the values of the data

$$r = \frac{1}{(n-1)} \sum \frac{(X-\mu_X)(Y-\mu_Y)}{\sigma_X \sigma_Y}$$

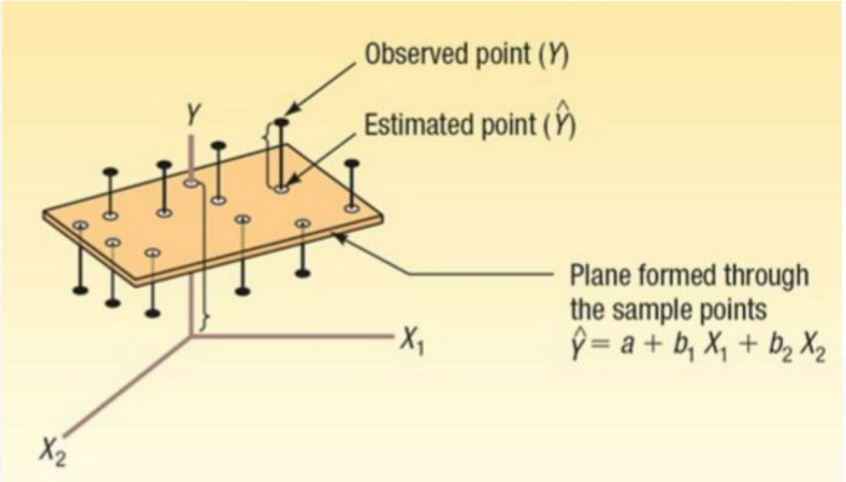
Linear Regression: Multiple Linear Regression

- Concepts in simple linear regression can be extended to multiple predictors
- Formula for n predictor variables

$$y = \beta_0 + \beta_1 x^1 + \beta_2 x^2 + \ldots + \beta_{n-1} x^{n-1} + \beta_n x^n$$

- Formula for 2 predictor variables:

$$y = \beta_0 + \beta_1 x^1 + \beta_2 x^2$$



Linear Regression: Standardization

The regression equation is simpler if variables are standardized so that their means are equal to 0 and standard deviations are equal to 1, for then $\beta_1 = r$ and $\beta_0 = 0$.

- Subtract by the mean
- Then, divide by the standard deviation

This makes the regression line:

$$Z_Y' = (r)(Z_X)$$

Where:

- Z_Y' - predicted standard score for Y
- r - correlation
- Z_X - standardized score for X

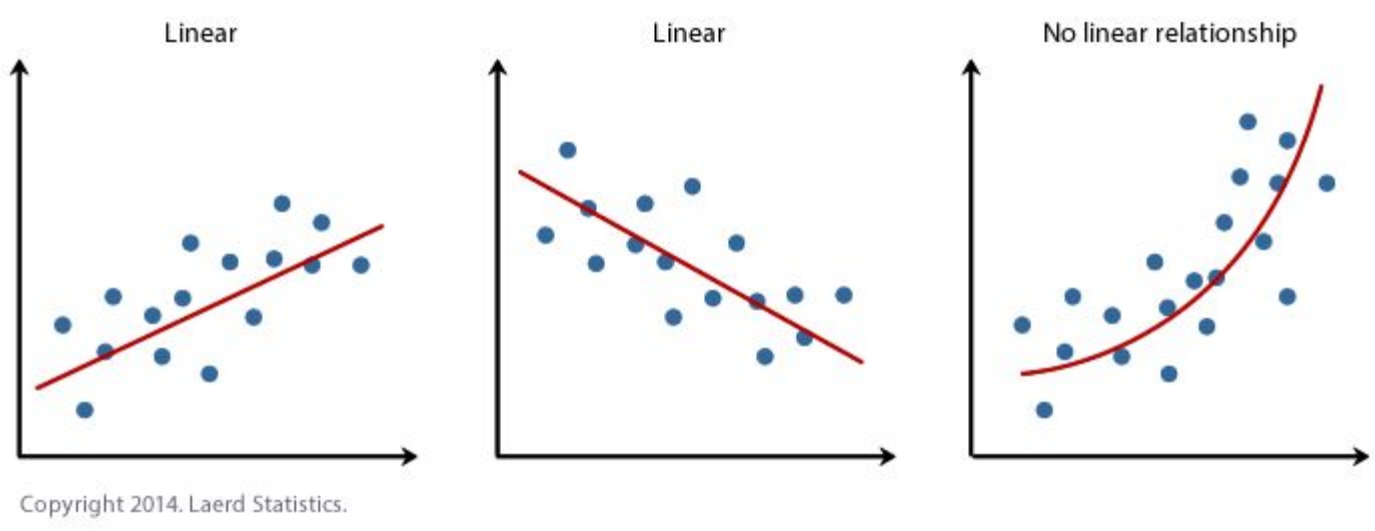
Note: the slope of the regression equation for standardized variables is r

Why standardize?
Simplifies the linear function (units of regression coefficients are the same)

Linear Regression: Key Requirements

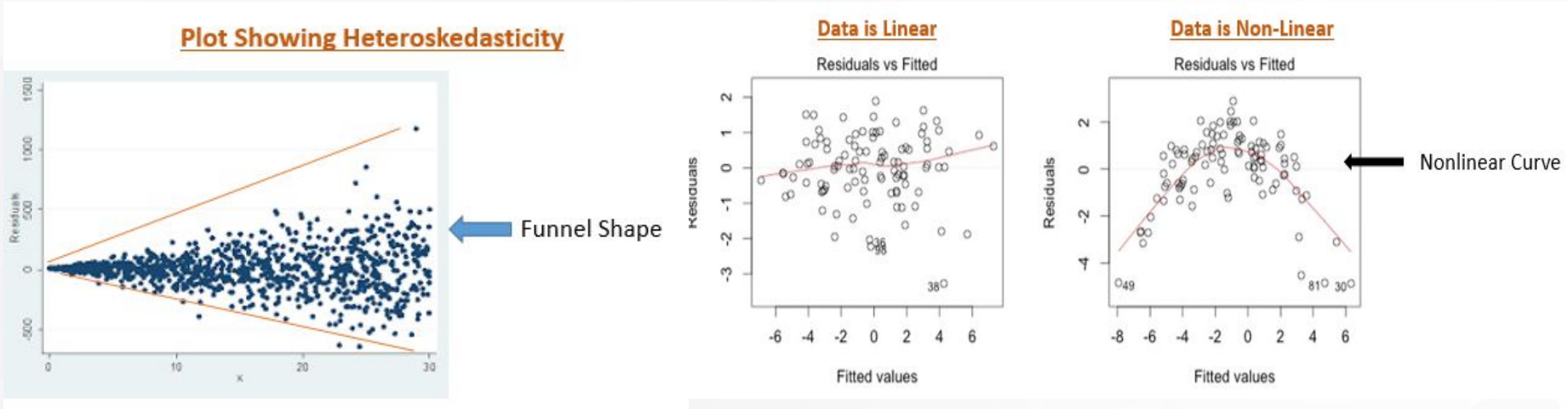
- Linear relationship
- Errors have the same variance (homoscedasticity)
- No or little high intercorrelations (multicollinearity) among the predictors
- Variables have normal distribution

Linear Regression Req't : Linear Relationship

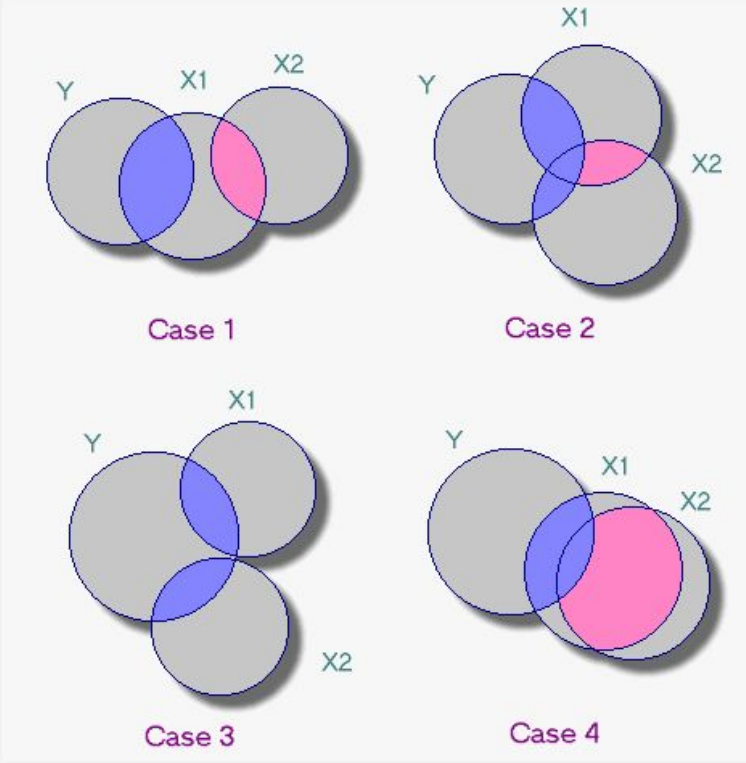


Linear Regression Req't : Homoscedasticity

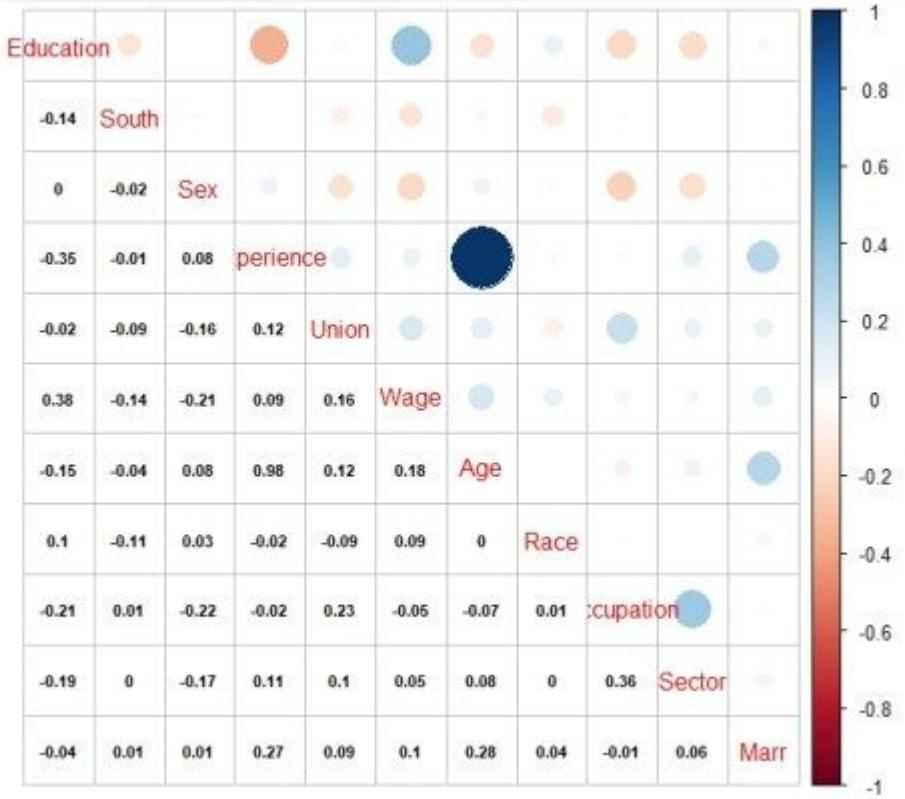
Errors have the same variance as you move along regression line. If residuals plot shows pattern, then there is heteroscedasticity and data is not fit for linear regression



No Multi-collinearity



Multi-collinearity : Correlation Plot



Linear Regression: Pros and Cons

+	-

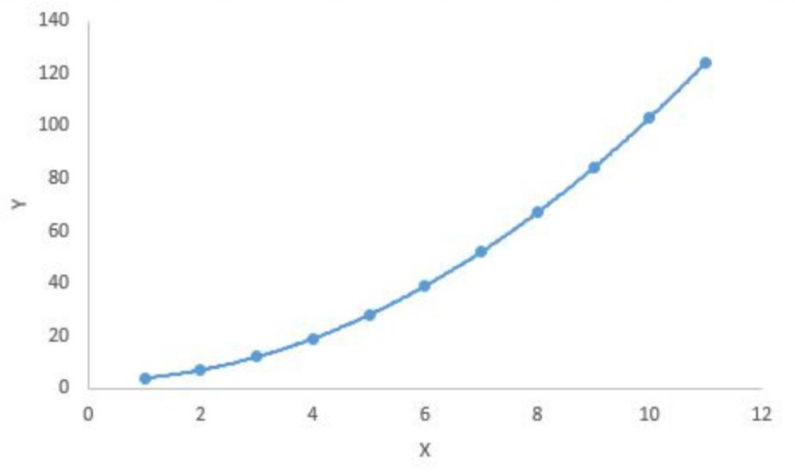
• Fast prediction
• Allows understanding of relationship between variables
• Works well if your data has a clear linear trend
	• Often inappropriately used to model non-linear relationships
	• Only looks at the mean of the target variable
	• Sensitive to outliers
	  ○ Normalize
	  ○ z-scores - remove those below -3.29 & above 3.29
	• Data must be independent

LAB:
Linear & Multiple Linear Regression

Analytiks

Polynomial Regression: Overview

- Fits the data into a regression line/curve using a polynomial equation
    - Exponent of predictor variable >1
    - $y = \beta_0 + \beta_1 x^2$
    - $y = \beta_0 + \beta_1 x^2 + \beta_1 x^3 + ... + \beta_1 x^n$
- Prediction: plug in the X values into the function
- Compared to Linear Regression, the best-fitting line in polynomial regression is a curve rather than a straight line



Polynomial Regression: Implementation

- A simple linear regression can be extended by constructing polynomial features from the coefficients
- In the standard linear regression case:

$$\hat{y}(w, x) = w_0 + w_1 x_1 + w_2 x_2$$

- If we want to fit a paraboloid to the data instead of a plane, we can combine the features in second-order polynomials, so that the model looks like this:

$$\hat{y}(w, x) = w_0 + w_1 x_1 + w_2 x_2 + w_3 x_1 x_2 + w_4 x_1^2 + w_5 x_2^2$$

- This is still a linear model! Create a new variable z:

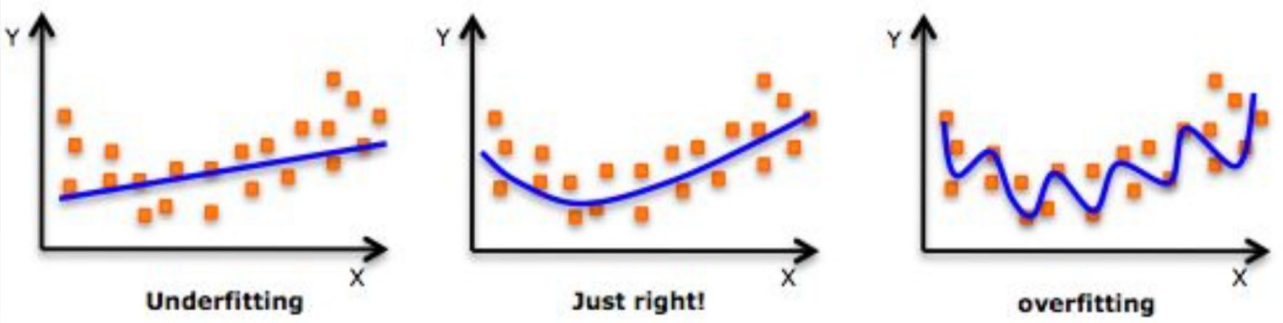
$$z = [x_1, x_2, x_1 x_2, x_1^2, x_2^2]$$

$$\hat{y}(w, x) = w_0 + w_1 z_1 + w_2 z_2 + w_3 z_3 + w_4 z_4 + w_5 z_5$$

- In Python, this is done using PolynomialFeatures pre-processor

Polynomial Regression: Overfitting

- There might be a temptation to fit a higher degree polynomial to get lower error
- Can result in overfitting
- Plot the relationships to see the fit and focus on making sure that the curve fits the nature of the problem
- Tip: keep the degree low (< 3)



Polynomial Regression: Pros and Cons

+	-
• Fast prediction • Allows understanding of relationship between variables • Have moderate flexibility of shapes • Useful when curvilinear effect is present in the dataset	• Finds polynomial relationship within the dataset • Poor interpolatory and extrapolatory properties • Sensitive to outliers

Regularization

Regression

Regularization

- Purpose: reduce overfitting
- Weights are penalized for growing too large
- Balance weights among different features
- Common regularization methods:
    - Ridge (L2) Regularization
    - Lasso (L1) Regularization

Regularization: Ridge Regression (L2)

- Ridge Regression (L2) is a technique for analyzing multiple regression data that suffer from multicollinearity.

- When multicollinearity occurs, least squares estimates are unbiased, but their variances are large so they may be far from the true value.

- Assumptions of this regression
  - Same as linear/polynomial regression
  - Linear/Polynomial relation exists
  - No multicollinearity
  - Except normality is NOT to be assumed

- Ridge regularization shrinks the value of coefficients but doesn't reach zero, which suggests no feature selection

Regularization: Ridge Regression (L2)

Regression algorithm combined with L2 regularization (a.k.a., ridge)

Objective = RSS + α * (sum of square of coefficients)

L2 norm: $\|\boldsymbol{x}\| := \sqrt{x_1^2 + \cdots + x_n^2}.$

$$\hat{\beta}^{\text{ridge}} = \underset{\beta \in \mathbb{R}^p}{\operatorname{argmin}} \sum_{i=1}^{n} (y_i - x_i^T \beta)^2 + \lambda \sum_{j=1}^{p} \beta_j^2$$

$$= \underset{\beta \in \mathbb{R}^p}{\operatorname{argmin}} \underbrace{\|y - X\beta\|_2^2}_{\text{Loss}} + \lambda \underbrace{\|\beta\|_2^2}_{\text{Penalty}}$$

1st part: least square term (i.e., prediction error)

2nd part: lambda of the summation of β2

Regularization: Lasso Regression (L1)

- Regression algorithm combined with L1 regularization (a.k.a., lasso)
- Least Absolute Shrinkage and Selection Operator or LASSO (L1) is a regression analysis method that performs both variable selection and regularization in order to enhance the prediction accuracy and interpretability of the statistical model it produces.
- Penalizes the absolute size of the regression coefficients
- Differs from ridge regression: uses absolute values in the penalty function, instead of squares
  - Causes some of the parameter estimates to turn out exactly zero
  - The larger the penalty applied, further the estimates get shrunk towards absolute zero. This results to variable selection out of given n variables.
- Assumptions of this regression
  - Same as linear/polynomial regression except normality is NOT to be assumed
- It shrinks coefficients to zero (exactly zero), which certainly helps in feature selection
  - If group of predictors are highly correlated, lasso picks only one of them and shrinks the others to zero

Regularization: Lasso Regression (L1)

Objective = RSS + α * (sum of absolute value of coefficients)

Uses a shrinkage parameter ƛ:

$$= \underset{\beta \in \mathbf{R}^p}{\operatorname{argmin}} \underbrace{\|y - X\beta\|_2^2}_{\text{Loss}} + \lambda \underbrace{\|\beta\|_1}_{\text{Penalty}}$$

1st part: least square term (i.e., prediction error)

2nd part: lambda of the summation of |β|

$$\lambda \sum_{j=0}^{p} |w_j|$$

LAB:
Polynomial Regression and Regularization

Analytiks

Metrics

Regression

Metrics

- R-squared
- Mean Absolute Error
- Weighted Mean Absolute Error
- Root Mean Squared Error

Metrics: R-Squared

- Coefficient of Determination

$$R2 = \frac{\text{Explained variation}}{\text{Total variation}}$$

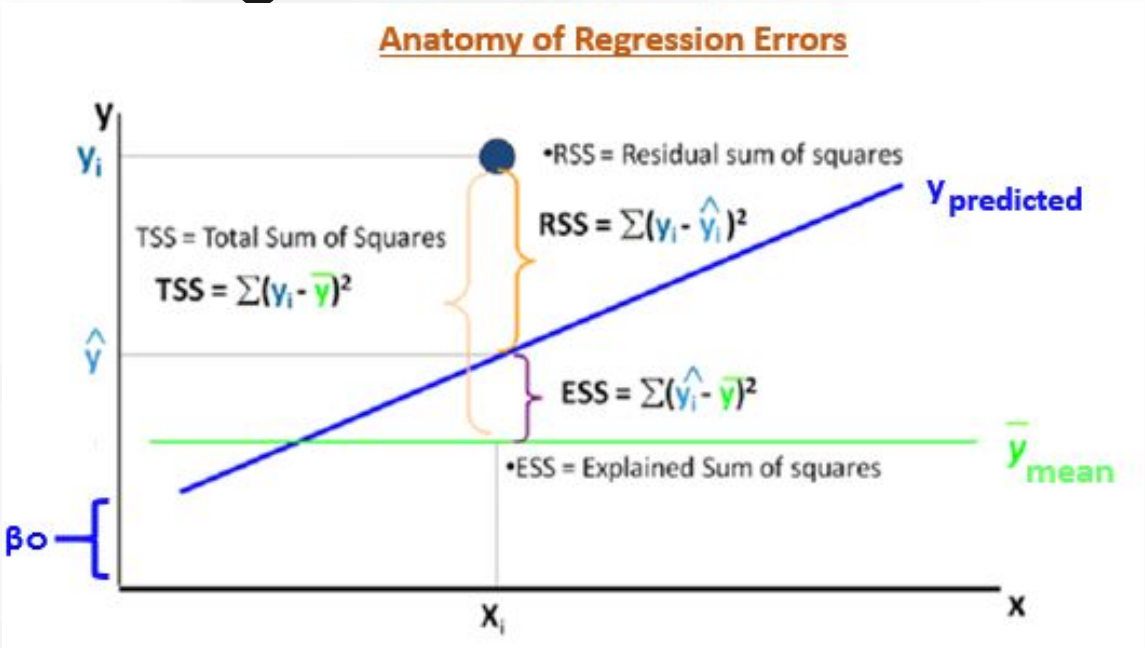
- % of the Variance explained by the model

- Between 0 to 100%
  - 0% - none of the variability of the response data around its mean.
  - 100% - all the variability of the response data around its mean.

$$R^2(y, \hat{y}) = 1 - \frac{\sum_{i=0}^{n_{\text{samples}}-1}(y_i - \hat{y}_i)^2}{\sum_{i=0}^{n_{\text{samples}}-1}(y_i - \bar{y})^2}$$

where $\bar{y} = \frac{1}{n_{\text{samples}}}\sum_{i=0}^{n_{\text{samples}}-1} y_i$

Understanding R2

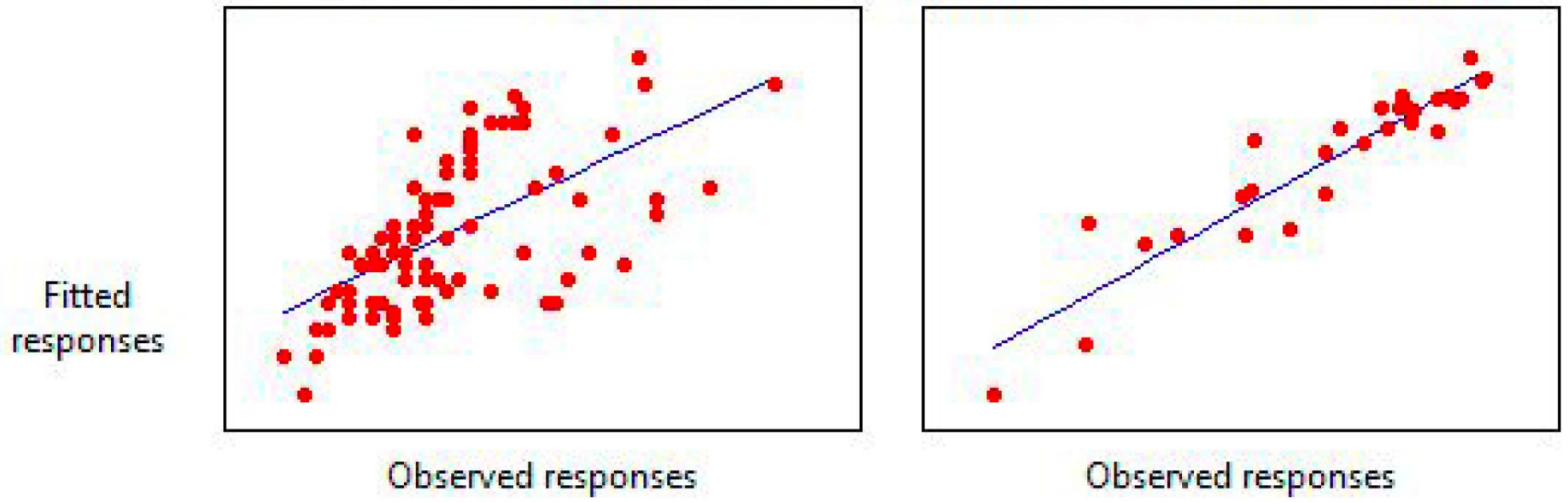


$$R^2 = \frac{ESS}{TSS}$$

$$R^2 = 1 - \frac{RSS}{TSS}$$

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y}_i)^2}$$

Which of the 2 have a higher R2?



Metrics: Mean Absolute Error (MAE)

Mean Absolute Error (MAE) is a quantity used to measure how close forecasts or predictions are to the eventual outcomes. The mean absolute error is given by:

$$\mathrm{MAE} = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i| = \frac{1}{n}\sum_{i=1}^{n}|e_i|$$

Where:

- AE = $|e_i| = |y_i - \hat{y}_i|$
- Actual = y_i
- Predicted = $\hat{y}_i$

Metrics: Weighted Mean Absolute Error (MAE)

A weighting factor would indicate the subjective importance we wish to place on each prediction, relating the error to any feature that might be relevant from both, the user or the seller point of view.

The Weighted Mean Absolute Error can be computed as:

$$\mathrm{WMAE} = \frac{1}{n}\sum_{i=1}^{n} w_i \left| y_i - \hat{y}_i \right|$$

Where:
Actual = y_i
Predicted = $\hat{y}_i$
Weight = w_i

Metrics: Root Mean Squared Error (RMSE)

- Very commonly used
- An excellent general purpose error metric for numerical predictions
- Compared to MAE: RMSE amplifies and severely punishes large errors

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}$$

- Where:
  - Actual = y_i
  - Predicted = $\hat{y}_i$

HW: Regression

Analytiks