Week 6

ARTIFICIAL INTELLIGENCE AND MACHINE LEARNING

What is data splitting?

Data splitting is when data is divided into two or more subsets. Typically, with a two-part split, one part is used to evaluate or test the data and the other to train the model.

Data splitting is an important aspect of data science, particularly for creating models based on data. This technique helps ensure the creation of data models and processes that use data models -- such as machine learning -- are accurate.

How data splitting works

In a basic two-part data split, the training data set is used to train and develop models. Training sets are commonly used to estimate different parameters or to compare different model performance.

The testing data set is used after the training is done. The training and test data are compared to check that the final model works correctly. With machine learning, data is commonly split into three or more sets. With three sets, the additional set is the dev set, which is used to change learning process parameters.

There is no set guideline or metric for how the data should be split; it may depend on the size of the original data pool or the number of predictors in a predictive model.

Data splitting in machine learning

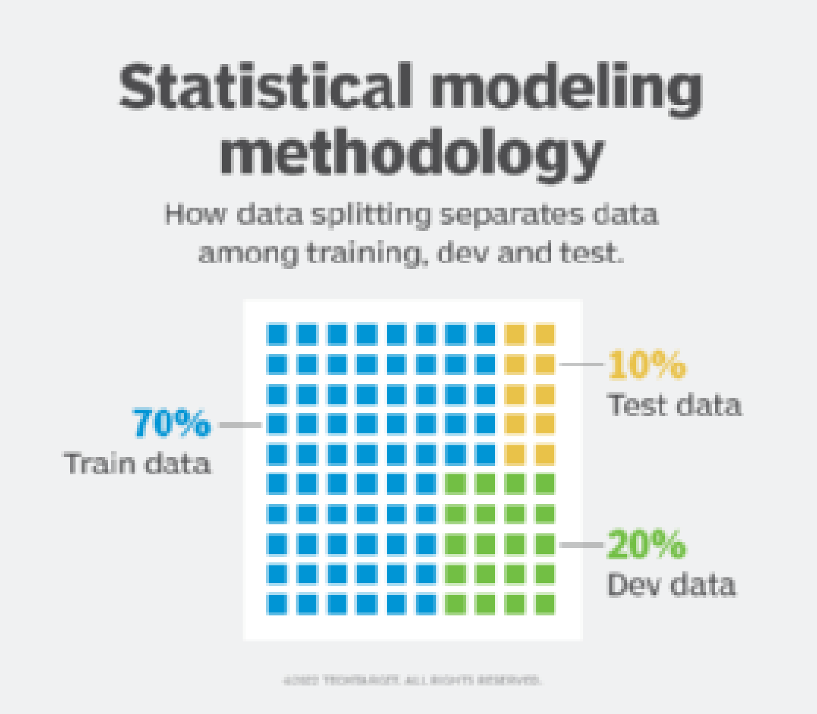
In machine learning, data splitting is typically done to avoid overfitting. That is an instance where a machine learning model fits its training data too well and fails to reliably fit additional data.

The original data in a machine learning model is typically taken and split into three or four sets. The three sets commonly used are the training set, the dev set and the testing set:

1. The **training set** is the portion of data used to train the model. The model should observe and learn from the training set, optimizing any of its parameters.

2. The **dev set** is a data set of examples used to change learning process parameters. It is also called the *cross-validation* or *model* validation set. This set of data has the goal of ranking the model's accuracy and can help with model selection.

3. The **testing set** is the portion of data that is tested in the final model and is compared against the previous sets of data. The testing set acts as an evaluation of the final mode and algorithm.



Data splitting separates data among train, test and dev data.

Data should be split so that data sets can have a high amount of training data. For example, data might be split at an 80-20 or a 70-30 ratio of training vs. testing data. The exact ratio depends on the data, but a 70-20-10 ratio for training, dev and test splits is optimal for small data sets.

Underfitting: A statistical model or a machine learning algorithm is said to have underfitting when it cannot capture the underlying trend of the data, i.e., it only performs well on training data but performs poorly on testing data. (It’s just like trying to fit undersized pants!) Underfitting destroys the accuracy of our machine learning model. Its occurrence simply means that our model or the algorithm does not fit the data well enough. It usually happens when we have fewer data to build an accurate model and also when we try to build a linear model with fewer non-linear data. In such cases, the rules of the machine learning model are too easy and flexible to be applied to such minimal data and therefore the model will probably make a lot of wrong predictions. Underfitting can be avoided by using more data and also reducing the features by feature selection.

In a nutshell, Underfitting refers to a model that can neither performs well on the training data nor generalize to new data.

Reasons for Underfitting:

1. High bias and low variance
2. The size of the training dataset used is not enough.
3. The model is too simple.
4. Training data is not cleaned and also contains noise in it.

Techniques to reduce underfitting:

1. Increase model complexity
2. Increase the number of features, performing feature engineering
3. Remove noise from the data.
4. Increase the number of epochs or increase the duration of training to get better results.

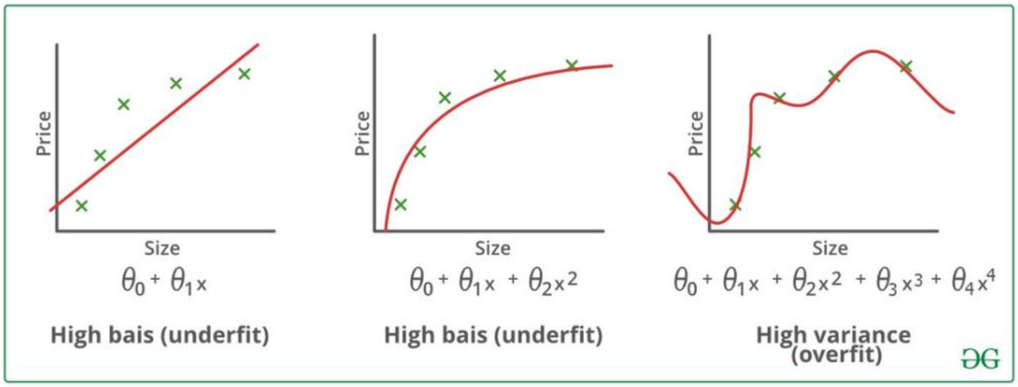
Overfitting: A statistical model is said to be overfitted when the model does not make accurate predictions on testing data. When a model gets trained with so much data, it starts learning from the noise and inaccurate data entries in our data set. And when testing with test data results in High variance. Then the model does not categorize the data correctly, because of too many details and noise. The causes of overfitting are the non-parametric and non-linear methods because these types of machine learning algorithms have more freedom in building the model based on the dataset and therefore they can really build unrealistic models. A solution to avoid overfitting is using a linear algorithm if we have linear data or using the parameters like the maximal depth if we are using decision trees.

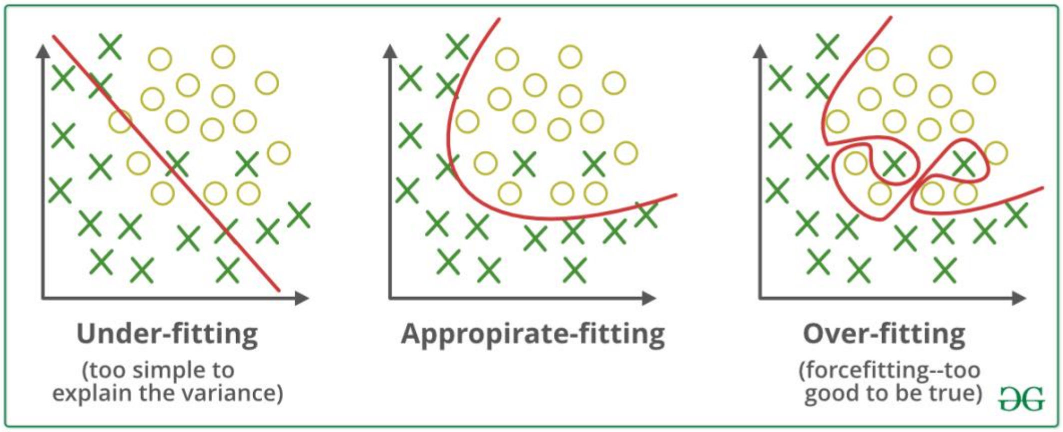
In a nutshell, Overfitting is a problem where the evaluation of machine learning algorithms on training data is different from unseen data.

Reasons for Overfitting are as follows:

1. High variance and low bias
2. The model is too complex
3. The size of the training data

Examples:




Under-fitting
(too simple to explain the variance)
Appropirate-fitting
Over-fitting
(forcefitting--too good to be true)

Techniques to reduce overfitting:

1. Increase training data.
2. Reduce model complexity.
3. Early stopping during the training phase (have an eye over the loss over the training period as soon as loss begins to increase stop training).
4. Ridge Regularization and Lasso Regularization
5. Use dropout for neural networks to tackle overfitting.

Machine Learning pipeline:

Model training

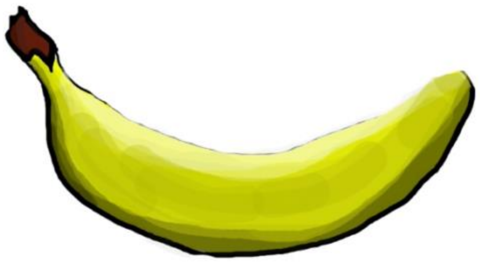
What is supervised learning?

Supervised learning, as the name indicates, has the presence of a supervisor as a teacher. Basically supervised learning is when we teach or train the machine using data that is well labelled. Which means some data is already tagged with the correct answer. After that, the machine is provided with a new set of examples(data) so that the supervised learning algorithm analyses the training data(set of training examples) and produces a correct outcome from labelled data.

For instance, suppose you are given a basket filled with different kinds of fruits. Now the first step is to train the machine with all the different fruits one by one like this:

- If the shape of the object is rounded and has a depression at the top, is red in color, then it will be labeled as –**Apple**.
- If the shape of the object is a long curving cylinder having Green-Yellow color, then it will be labeled as –**Banana**.

Now suppose after training the data, you have given a new separate fruit, say Banana from the basket, and asked to identify it.



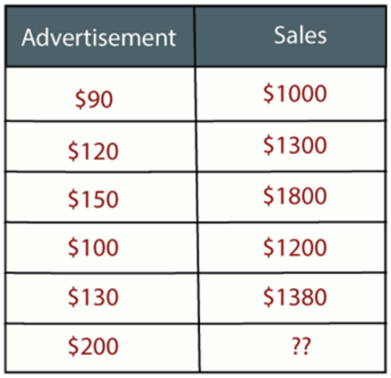
Since the machine has already learned the things from previous data and this time has to use it wisely. It will first classify the fruit with its shape and color and would confirm the fruit name as BANANA and put it in the Banana category. Thus the machine learns the things from training data(basket containing fruits) and then applies the knowledge to test data(new fruit).

Regression Analysis in Machine learning

Regression analysis is a statistical method to model the relationship between a dependent (target) and independent (predictor) variables with one or more independent variables. More specifically, Regression analysis helps us to understand how the value of the dependent variable is changing corresponding to an independent variable when other independent variables are held fixed. It predicts continuous/real values such as **temperature, age, salary, price,** etc.

We can understand the concept of regression analysis using the below example:

Example: Suppose there is a marketing company A, who does various advertisement every year and get sales on that. The below list shows the advertisement made by the company in the last 5 years and the corresponding sales:



Advertisement	Sales
$90	$1000
$120	$1300
$150	$1800
$100	$1200
$130	$1380
$200	??

Now, the company wants to do the advertisement of $200 in the year 2019 **and wants to know the prediction about the sales for this year**. So to solve such type of prediction problems in machine learning, we need regression analysis.

Types of Regression Analysis Techniques

There are many types of regression analysis techniques, and the use of each method depends upon the number of factors. These factors include the type of target variable, shape of the regression line, and the number of independent variables.

Below are the different regression techniques:

1. Linear Regression
2. Logistic Regression
3. Ridge Regression
4. Lasso Regression
5. Polynomial Regression
6. Bayesian Linear Regression

The different types of regression in machine learning techniques are explained below in detail:

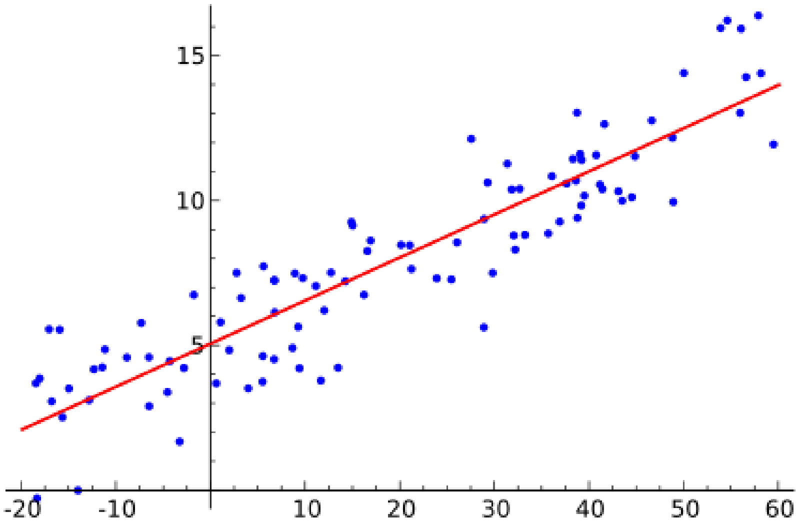
1. Linear Regression

Linear regression is one of the most basic types of regression in machine learning. The linear regression model consists of a predictor variable and a dependent variable related linearly to each other. In case the data involves more than one independent variable, then linear regression is called multiple linear regression models.

The below-given equation is used to denote the linear regression model:

y=mx+c+e

where m is the slope of the line, c is an intercept, and e represents the error in the model.



Source

The best fit line is determined by varying the values of m and c. The predictor error is the difference between the observed values and the predicted value. The values of m and c get selected in such a way that it gives the minimum predictor error. It is important to note that a simple linear regression model is susceptible to outliers. Therefore, it should not be used in case of big size data.

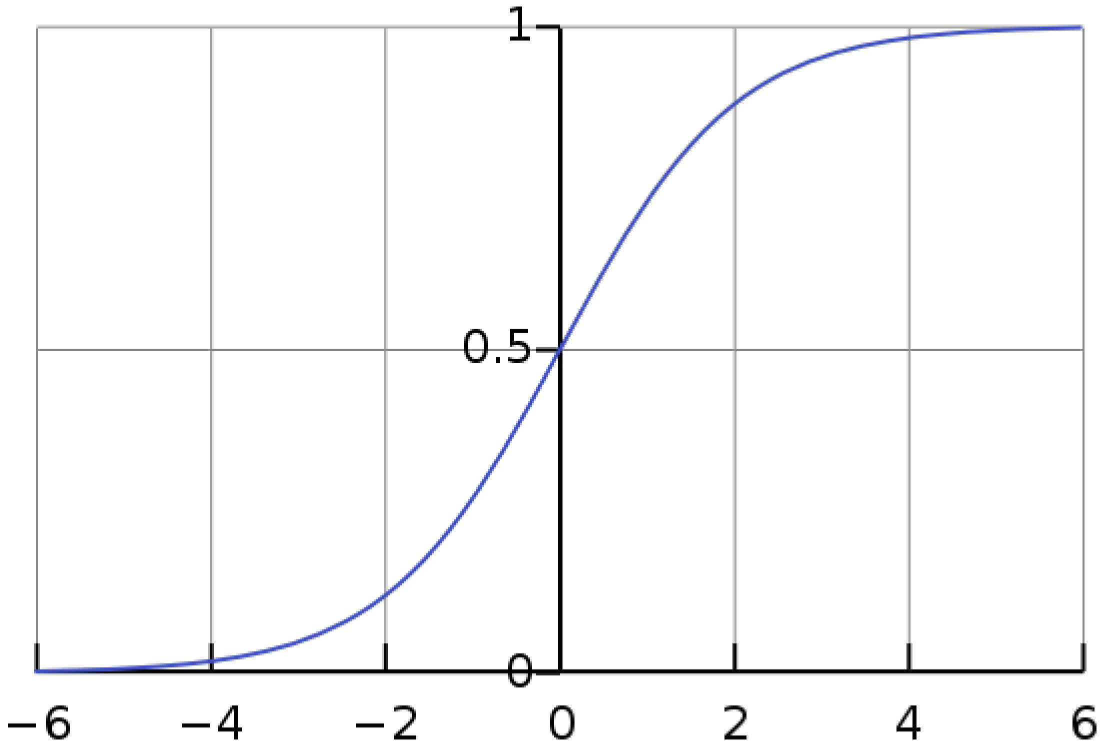
2. Logistic Regression

Logistic regression is one of the types of regression analysis technique, which gets used when the dependent variable is discrete. Example: 0 or 1, true or false, etc. This means the target variable can have only two values, and a sigmoid curve denotes the relation between the target variable and the independent variable.

Logit function is used in Logistic Regression to measure the relationship between the target variable and independent variables. Below is the equation that denotes the logistic regression.

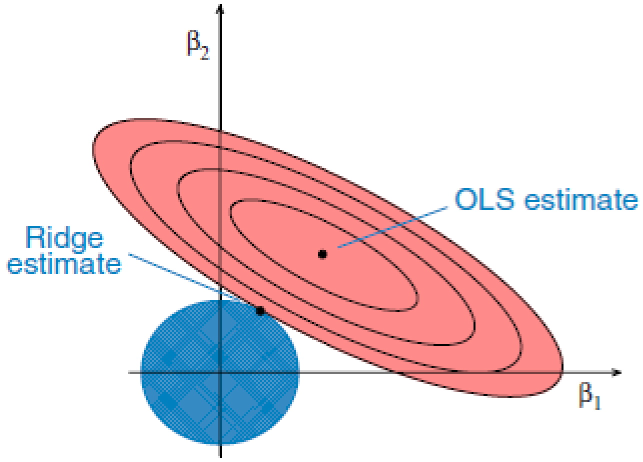
logit(p) = ln(p/(1-p)) = b0+b1X1+b2X2+b3X3….+bkXk

where p is the probability of occurrence of the feature.



For selecting logistic regression, as the regression analyst technique, it should be noted, the size of data is large with the almost equal occurrence of values to come in target variables. Also, there should be no multicollinearity, which means that there should be no correlation between independent variables in the dataset.

3. Ridge Regression

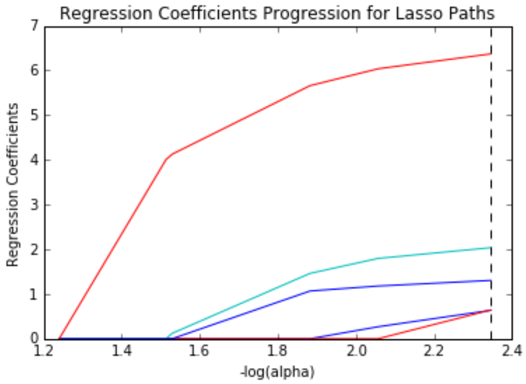


This is another one of the types of regression in machine learning which is usually used when there is a high correlation between the independent variables. This is because, in the case of multi collinear data, the least square estimates give unbiased values. But, in case the collinearity is very high, there can be some bias value. Therefore, a bias matrix is introduced in the equation of Ridge Regression. This is a powerful regression method where the model is less susceptible to overfitting.

4. Lasso Regression

Lasso Regression is one of the types of regression in machine learning that performs regularization along with feature selection. It prohibits the absolute size of the regression coefficient. As a result, the coefficient value gets nearer to zero, which does not happen in the case of Ridge Regression.

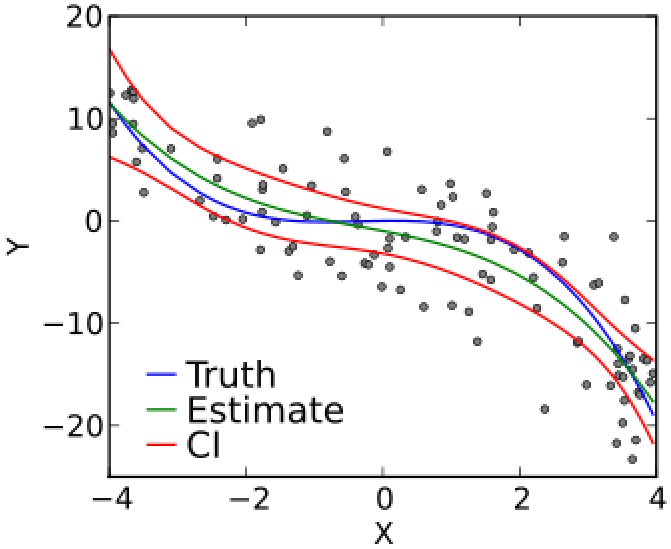
Due to this, feature selection gets used in Lasso Regression, which allows selecting a set of features from the dataset to build the model. In the case of Lasso Regression, only the required features are used, and the other ones are made zero. This helps in avoiding the overfitting in the model. In case the independent variables are highly collinear, then Lasso regression picks only one variable and makes other variables to shrink to zero.



5. Polynomial Regression

Polynomial Regression is another one of the types of regression analysis techniques in machine learning, which is the same as Multiple Linear Regression with a little modification. In Polynomial Regression, the relationship between independent and dependent variables, that is X and Y, is denoted by the n-th degree.

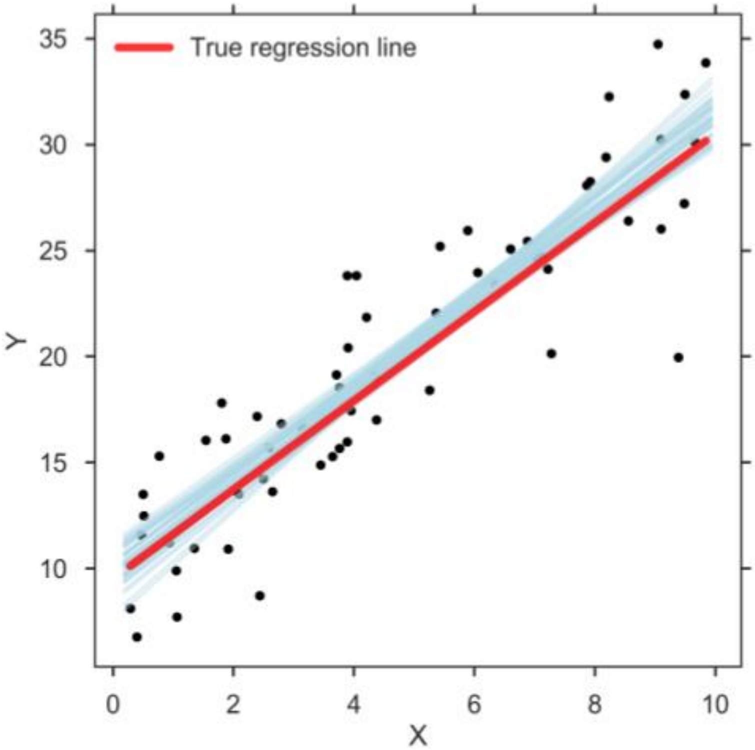
It is a linear model as an estimator. Least Mean Squared Method is used in Polynomial Regression also. The best fit line in Polynomial Regression that passes through all the data points is not a straight line, but a curved line, which depends upon the power of X or value of n.



While trying to reduce the Mean Squared Error to a minimum and to get the best fit line, the model can be prone to overfitting. It is recommended to analyze the curve towards the end as the higher Polynomials can give strange results on extrapolation.

6. Bayesian Linear Regression

Bayesian Regression is one of the types of regression in machine learning that uses the Bayes theorem to find out the value of regression coefficients. In this method of regression, the posterior distribution of the features is determined instead of finding the least-squares. Bayesian Linear Regression is like both Linear Regression and Ridge Regression but is more stable than the simple Linear Regression.



Regression plays a vital role in predictive modelling and is found in many machine learning applications. Algorithms from the regressions provide different perspectives regarding the relationship between the variables and their outcomes. These set models could then be used as a guideline for fresh input data or to find missing data.

As the models are trained to understand a variety of relationships between different variables, they are often extremely helpful in predicting the portfolio performance or stocks and trends. These implementations fall under machine learning in finance.

The very common use of regression in AI includes:

- Predicting a company's sales or marketing success
- Generating continuous outcomes like stock prices
- Forecasting different trends or customer's purchase behavior

What is Regularization in Machine Learning?

Regularization refers to techniques that are used to calibrate machine learning models in order to minimize the adjusted loss function and prevent overfitting or underfitting.

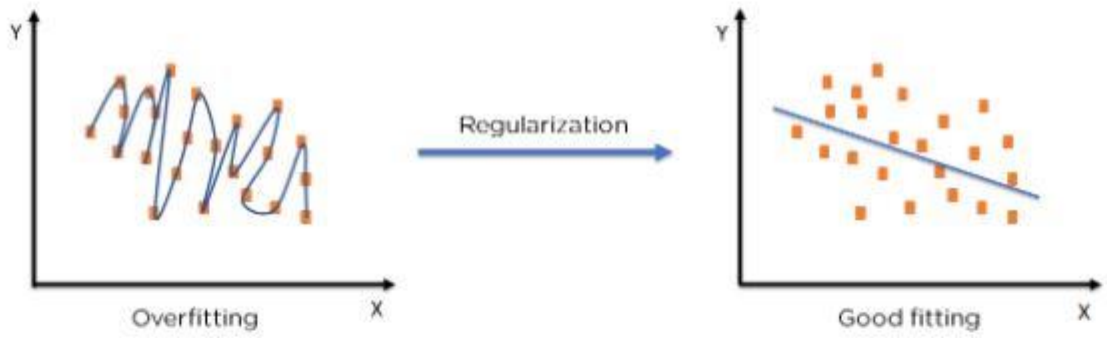


Figure 5: Regularization on an over-fitted model

Using Regularization, we can fit our machine learning model appropriately on a given test set and hence reduce the errors in it.

Regularization Techniques

There are two main types of regularization techniques: Ridge Regularization and Lasso Regularization.

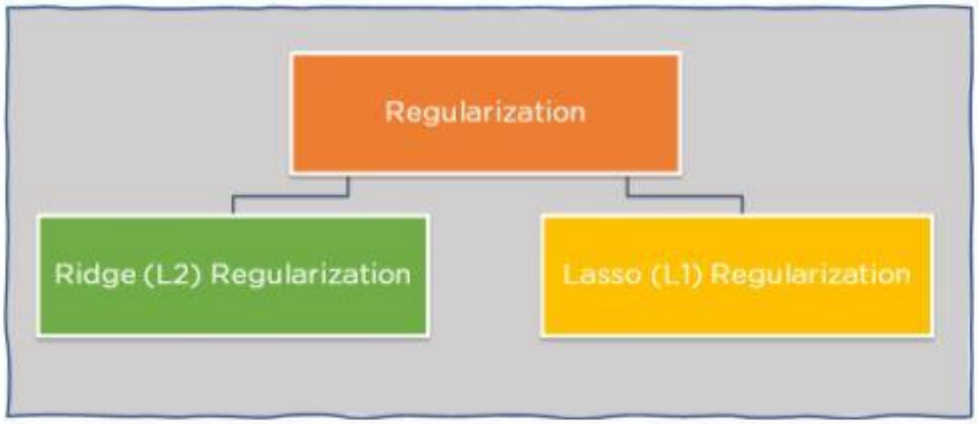
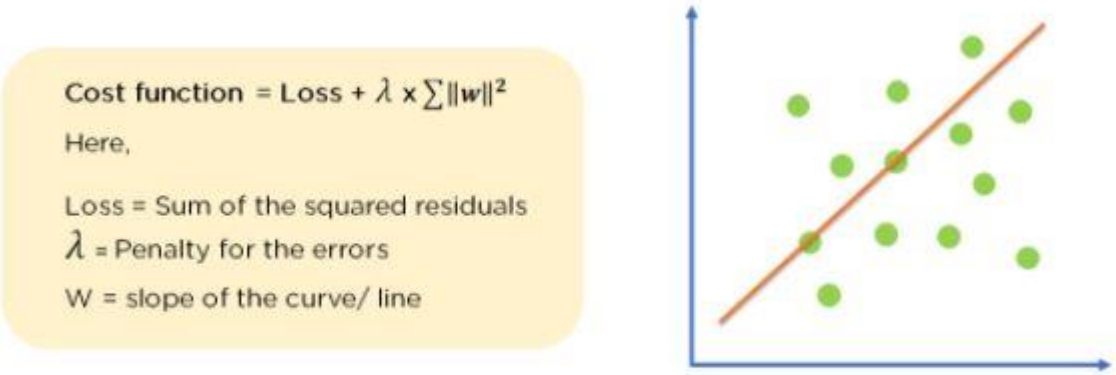


Figure 6: Regularization techniques

Ridge Regularization :

Also known as Ridge Regression, it modifies the over-fitted or under fitted models by adding the penalty equivalent to the sum of the squares of the magnitude of coefficients.

This means that the mathematical function representing our machine learning model is minimized and coefficients are calculated. The magnitude of coefficients is squared and added. Ridge Regression performs regularization by shrinking the coefficients present. The function depicted below shows the cost function of ridge regression :



Cost function = Loss + λ x $\sum \|w\|^2$

Here,

Loss = Sum of the squared residuals

λ = Penalty for the errors

W = slope of the curve/ line

Figure 7: Cost Function of Ridge Regression

In the cost function, the penalty term is represented by Lambda λ. By changing the values of the penalty function, we are controlling the penalty term. The higher the penalty, it reduces the magnitude of coefficients. It shrinks the parameters. Therefore, it is used to prevent multicollinearity, and it reduces the model complexity by coefficient shrinkage.

Consider the graph illustrated below which represents Linear regression :

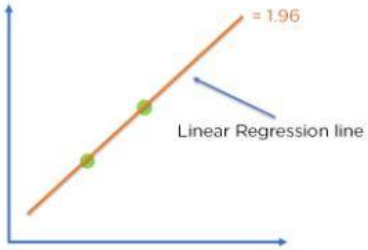


Figure 8: Linear regression model

Cost function = Loss + λ x∑‖w‖^2

For Linear Regression line, let’s consider two points that are on the line,

Loss = 0 (considering the two points on the line)

λ= 1

w = 1.4

Then, Cost function = 0 + 1 x 1.42

= 1.96

For Ridge Regression, let’s assume,

Loss = 0.32 + 0.22 = 0.13

λ = 1

w = 0.7

Then, Cost function = 0.13 + 1 x 0.72

= 0.62

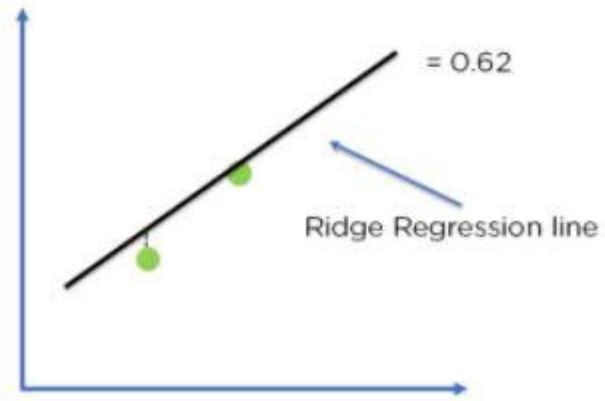


Figure 9: Ridge regression model

Comparing the two models, with all data points, we can see that the Ridge regression line fits the model more accurately than the linear regression line.

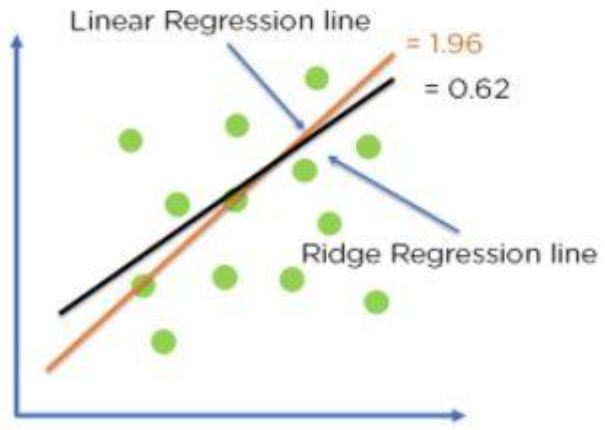
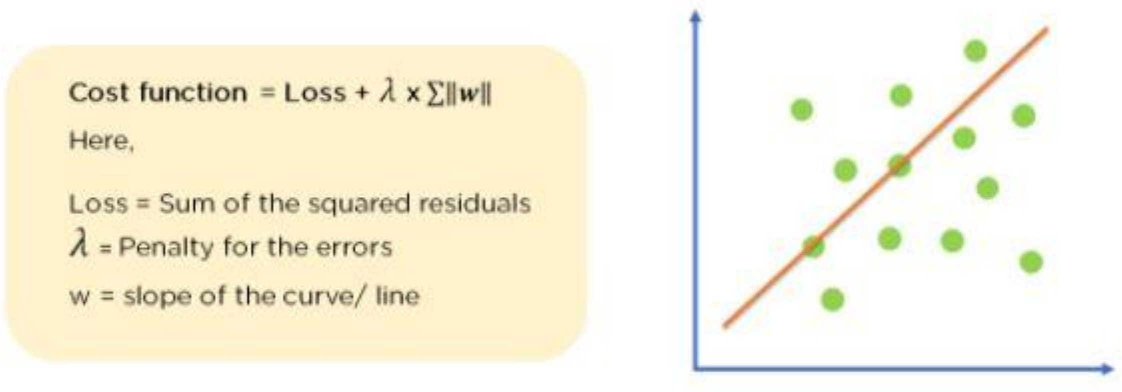


Figure 10: Optimization of model fit using Ridge Regression

Lasso Regression

It modifies the over-fitted or under-fitted models by adding the penalty equivalent to the sum of the absolute values of coefficients.

Lasso regression also performs coefficient minimization, but instead of squaring the magnitudes of the coefficients, it takes the true values of coefficients. This means that the coefficient sum can also be 0, because of the presence of negative coefficients. Consider the cost function for Lasso regression :



Cost function = **Loss** + λ x $\sum\|w\|$

Here,

Loss = Sum of the squared residuals

λ = Penalty for the errors

w = slope of the curve/ line

Figure 11: Cost function for Lasso Regression

We can control the coefficient values by controlling the penalty terms, just like we did in Ridge Regression. Again consider a Linear Regression model :

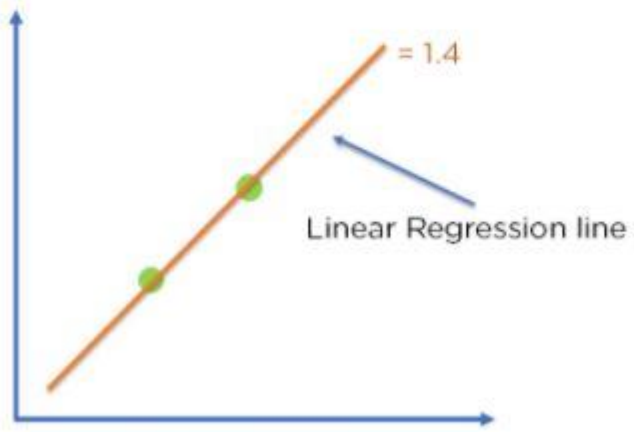


Figure 12: Linear Regression Model

Figure 12: Linear Regression Model

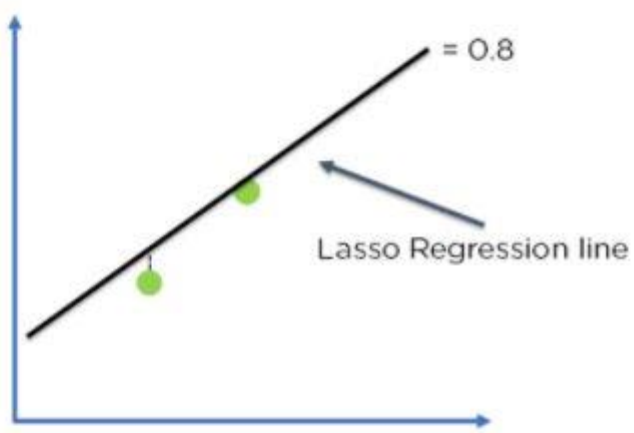


Figure 13: Lasso Regression

Comparing the two models, with all data points, we can see that the Lasso regression line fits the model more accurately than the linear regression line.

4 Examples of Using Linear Regression in Real Life

Linear regression is one of the most commonly used techniques in statistics. It is used to quantify the relationship between one or more predictor variables and a response variable.

The most basic form of linear is regression is known as simple linear regression, which is used to quantify the relationship between one predictor variable and one response variable.

If we have more than one predictor variable then we can use multiple linear regression, which is used to quantify the relationship between several predictor variables and a response variable.

This tutorial shares four different examples of when linear regression is used in real life.

Linear Regression Real Life Example #1

Businesses often use linear regression to understand the relationship between advertising spending and revenue.

For example, they might fit a simple linear regression model using advertising spending as the predictor variable and revenue as the response variable. The regression model would take the following form:

revenue = β_0 + β_1(ad spending)

The coefficient $\boldsymbol{\beta_0}$ would represent total expected revenue when ad spending is zero.

The coefficient $\boldsymbol{\beta_1}$ would represent the average change in total revenue when ad spending is increased by one unit (e.g. one dollar).

If β_1 is negative, it would mean that more ad spending is associated with less revenue.

If β_1 is close to zero, it would mean that ad spending has little effect on revenue.

And if β_1 is positive, it would mean more ad spending is associated with more revenue.

Depending on the value of β_1, a company may decide to either decrease or increase their ad spending.

Linear Regression Real Life Example #2

Medical researchers often use linear regression to understand the relationship between drug dosage and blood pressure of patients.

For example, researchers might administer various dosages of a certain drug to patients and observe how their blood pressure responds. They might fit a simple linear regression model using dosage as the predictor variable and blood pressure as the response variable. The regression model would take the following form:

blood pressure = β_0 + β_1(dosage)

The coefficient β_0 would represent the expected blood pressure when dosage is zero.

The coefficient β_1 would represent the average change in blood pressure when dosage is increased by one unit.

If β_1 is negative, it would mean that an increase in dosage is associated with a decrease in blood pressure.

If β_1 is close to zero, it would mean that an increase in dosage is associated with no change in blood pressure.

If β_1 is positive, it would mean that an increase in dosage is associated with an increase in blood pressure.

Depending on the value of β_1, researchers may decide to change the dosage given to a patient.

Linear Regression Real Life Example #3

Agricultural scientists often use linear regression to measure the effect of fertilizer and water on crop yields.

For example, scientists might use different amounts of fertilizer and water on different fields and see how it affects crop yield. They might fit a multiple linear regression model using fertilizer and water as the predictor variables and crop yield as the response variable. The regression model would take the following form:

crop yield = β_0 + β_1(amount of fertilizer) + β_2(amount of water)

The coefficient **β_0** would represent the expected crop yield with no fertilizer or water.

The coefficient **β_1** would represent the average change in crop yield when fertilizer is increased by one unit, *assuming the amount of water remains unchanged.*

The coefficient **β_2** would represent the average change in crop yield when water is increased by one unit, *assuming the amount of fertilizer remains unchanged.*

Depending on the values of β_1 and β_2, the scientists may change the amount of fertilizer and water used to maximize the crop yield.

Linear Regression Real Life Example #4

Data scientists for professional sports teams often use linear regression to measure the effect that different training regimens have on player performance.

For example, data scientists in the NBA might analyze how different amounts of weekly yoga sessions and weightlifting sessions affect the number of points a player scores. They might fit a multiple linear regression model using yoga sessions and weightlifting sessions as the predictor variables and total points scored as the response variable. The regression model would take the following form:

points scored = β_0 + β_1(yoga sessions) + β_2(weightlifting sessions)

The coefficient **β_0** would represent the expected points scored for a player who participates in zero yoga sessions and zero weightlifting sessions.

The coefficient **β_1** would represent the average change in points scored when weekly yoga sessions is increased by one, *assuming the number of weekly weightlifting sessions remains unchanged*.

The coefficient **β_2** would represent the average change in points scored when weekly weightlifting sessions is increased by one, *assuming the number of weekly yoga sessions remains unchanged*.

Depending on the values of β_1 and β_2, the data scientists may recommend that a player participates in more or less weekly yoga and weightlifting sessions in order to maximize their points scored.

Simple Linear Regression in Machine Learning

Simple Linear Regression is a type of Regression algorithms that models the relationship between a dependent variable and a single independent variable. The relationship shown by a Simple Linear Regression model is linear or a sloped straight line, hence it is called Simple Linear Regression.

The key point in Simple Linear Regression is that the ***dependent variable must be a continuous/real value***. However, the independent variable can be measured on continuous or categorical values.

Simple Linear regression algorithm has mainly two objectives:

- **Model the relationship between the two variables.** Such as the relationship between Income and expenditure, experience and Salary, etc.
- **Forecasting new observations.** Such as Weather forecasting according to temperature, Revenue of a company according to the investments in a year, etc.

Simple Linear Regression Model:

The Simple Linear Regression model can be represented using the below equation: $y= a_0+a_1x+ \varepsilon$

Where,

a0= It is the intercept of the Regression line (can be obtained putting x=0) a1= It is the slope of the regression line, which tells whether the line is increasing or decreasing.
ε = The error term. (For a good model it will be negligible)

What is Multiple Linear Regression in Machine Learning?

Linear regression is a model that predicts one variable's values based on another's importance. It's one of the most popular and widely-used models in machine learning, and it's also one of the first things you should learn as you explore machine learning.

Linear regression is so popular because it's so simple: all it does is try to predict values based on past data, which makes it easy to get started with and understand. The simplicity means it's also easy to implement, which makes it a great starting point if you're new to machine learning.

There are two types of linear regression algorithms -

- Simple - deals with two features.
- Multiple - deals with more than two features.

In this guide, let’s understand multiple linear regression in depth.

What Is Multiple Linear Regression (MLR)?

One of the most common types of predictive analysis is multiple linear regression. This type of analysis allows you to understand the relationship between a continuous dependent variable and two or more independent variables.

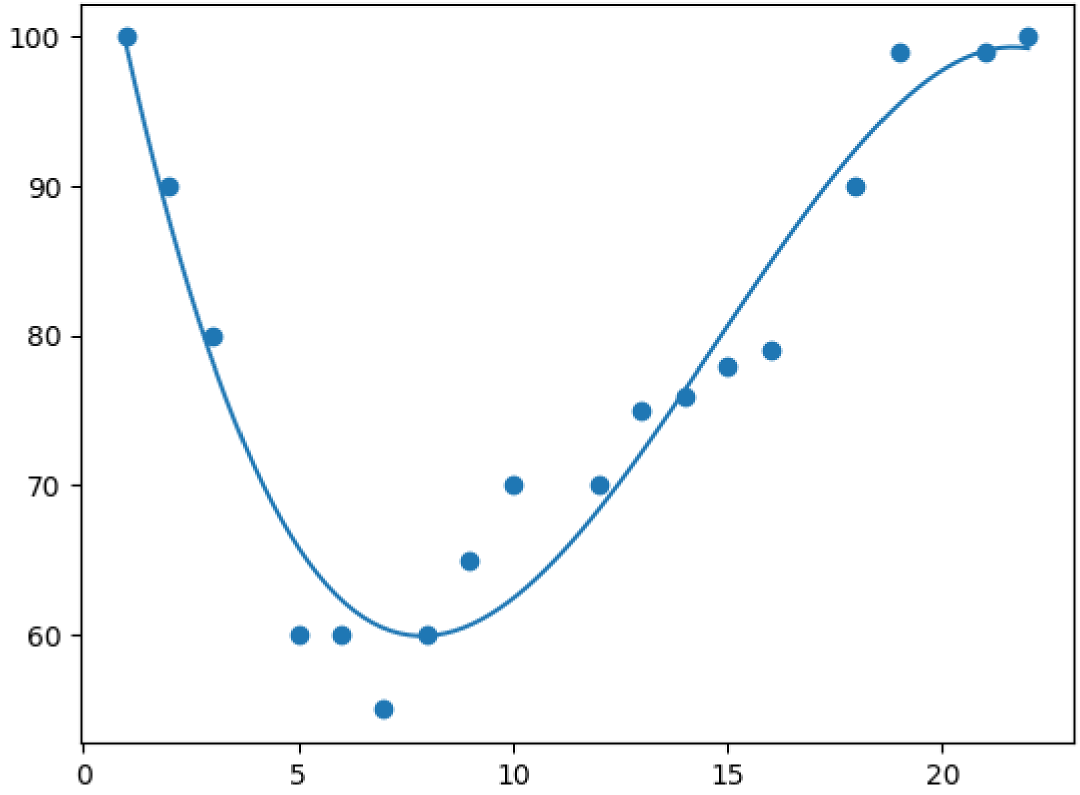
The independent variables can be either continuous (like age and height) or categorical (like gender and occupation). It's important to note that if your dependent variable is categorical, you should dummy code it before running the analysis.

- Polynomial linear regression

Polynomial Regression

If your data points clearly will not fit a linear regression (a straight line through all data points), it might be ideal for polynomial regression.

Polynomial regression, like linear regression, uses the relationship between the variables x and y to find the best way to draw a line through the data points.



How Does it Work?

Python has methods for finding a relationship between data-points and to draw a line of polynomial regression. We will show you how to use these methods instead of going through the mathematic formula.

In the example below, we have registered 18 cars as they were passing a certain tollbooth.

We have registered the car's speed, and the time of day (hour) the passing occurred.

The x-axis represents the hours of the day and the y-axis represents the speed:

Understanding Simple linear regression

- **Regression equation**

Simple linear regression formula

The formula for a simple linear regression is:

$$y = \beta_0 + \beta_1 X + \epsilon$$

- **y** is the predicted value of the dependent variable (**y**) for any given value of the independent variable (**x**).
- **B_0** is the **intercept**, the predicted value of **y** when the **x** is 0.
- **B_1** is the regression coefficient – how much we expect **y** to change as **x** increases.
- **x** is the independent variable (the variable we expect is influencing **y**).
- **e** is the **error** of the estimate, or how much variation there is in our estimate of the regression coefficient.

Linear regression finds the line of best fit line through your data by searching for the regression coefficient (B_1) that minimizes the total error (e) of the model.

While you can perform a linear regression by hand, this is a tedious process, so most people use statistical programs to help them quickly analyze the data.

Assumptions of simple linear regression

Simple linear regression is a **parametric test**, meaning that it makes certain assumptions about the data. These assumptions are:

1. **Homogeneity of variance (homoscedasticity)**: the size of the error in our prediction doesn't change significantly across the values of the independent variable.
2. **Independence of observations**: the observations in the dataset were collected using statistically valid sampling methods, and there are no hidden relationships among observations.
3. **Normality**: The data follows a normal distribution.

Linear regression makes one additional assumption:

4. The relationship between the independent and dependent variable is **linear**: the line of best fit through the data points is a straight line (rather than a curve or some sort of grouping factor).

If your data do not meet the assumptions of homoscedasticity or normality, you may be able to use a nonparametric test instead, such as the Spearman rank test.

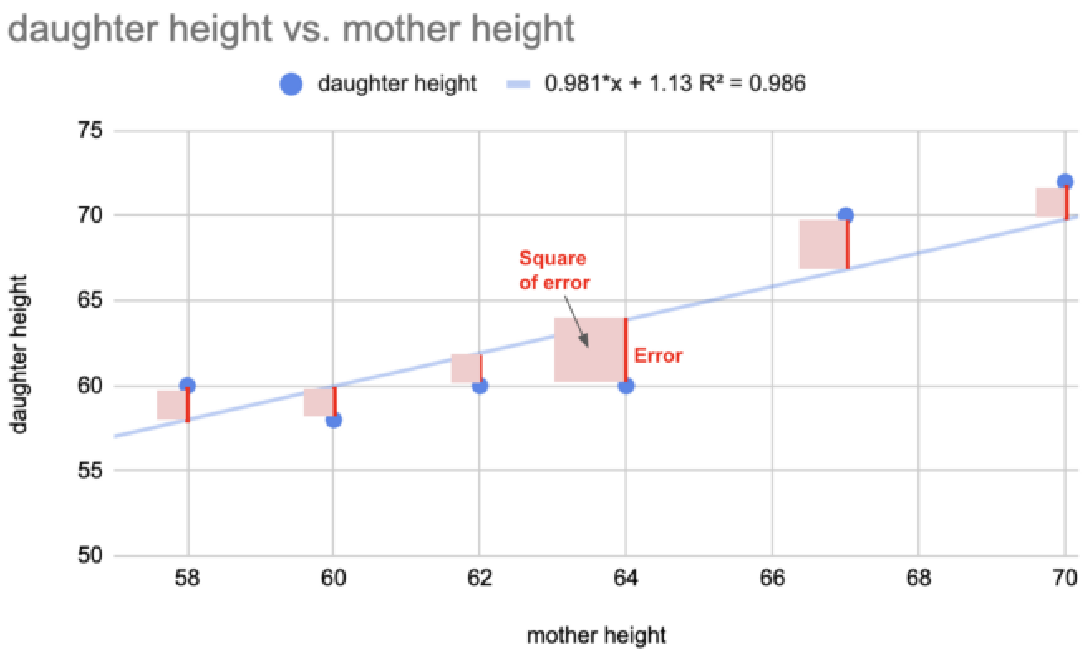
Example: Data that doesn't meet the assumptionsYou think there is a linear relationship between cured meat consumption and the incidence of colorectal cancer in the U.S. However, you find that much more data has been collected at high rates of meat consumption than at low rates of meat consumption, with the result that there is much more variation in the estimate of cancer rates at the low range than at the high range. Because the data violate the assumption of homoscedasticity, it doesn't work for regression, but you perform a Spearman rank test instead.
If your data violate the assumption of independence of observations (e.g., if observations are repeated over time), you may be able to perform a linear mixed-effects model that accounts for the additional structure in the data.

Gradient descent way of computing line of best fit:

In gradient descent, you start with a random line. Then you change the parameters of the line (i.e. slope and y-intercept) little by little to arrive at the line of best fit.

How do you know when you arrived at the line of best fit?

For every line you try — line A, line B, line C, etc — you calculate the sum of squares of the errors. If line B has a smaller value than line A, then line B is a better fit, etc.



Error and square of error

Error is your actual value minus your predicted value. The line of best fit minimizes the sum of the squares of all the errors. In linear regression, the line of best fit we computed above using correlation coefficient also happens to be the least squared error line. That's why the regression line is called the LEAST SQUARE REGRESSION LINE.

Model Evaluation & testing

Evaluate regression model: