**Linear Regression Types**

<https://www.analyticsvidhya.com/blog/2015/08/comprehensive-guide-regression/>

Linear and Logistic regression techniques are usually the first algorithms people learn in [data science](https://courses.analyticsvidhya.com/courses/introduction-to-data-science-2/?utm_source=blog&utm_medium=7regressiontypesarticle). Due to their popularity, a lot of analysts even end up thinking that they are the only form of regression techniques. The ones who are slightly more involved think that they are the most important among all forms of regression analysis.

The truth is that there are innumerable forms of regression methods. Each form holds its own importance and is best to apply in specific conditions. In this article, I have explained the most commonly used 7 types of regression in data science in a simple manner.

Through this article, I also hope that people develop an idea of the breadth of regressions, instead of just applying linear/logistic regression to every machine learning problem they come across and hoping that they would just fit!

If you’re new to data science and seeking a place to start your journey, we offer some comprehensive courses that might interest you:

* [Introduction to Data Science Course](https://courses.analyticsvidhya.com/courses/introduction-to-data-science-2/?utm_source=blog&utm_medium=7regressiontypesarticle): Covering the core topics of Python, Statistics and Predictive Modeling, it is the perfect way to take your first steps into data science
* [Certified AI & ML Blackbelt+ Program](https://courses.analyticsvidhya.com/bundles/certified-ai-ml-blackbelt-plus?utm_source=blog&utm_medium=7regressiontypesarticle)

Learning Objectives

* Familiarize yourself with the different regression types in [machine learning](https://www.analyticsvidhya.com/machine-learning/?utm_source=blog&utm_medium=7-regression-techniques), including linear and logistic regression.
* Learn the regression equation and regression coefficients of each type.
* Know to differentiate between the 7 types of regression techniques

**What Is Regression Analysis?**

Regression analysis is a form of predictive modelling technique which investigates the relationship between a **dependent**(target) and **independent variable (s)** (predictor). This technique is used for forecasting, time series modelling and finding the [causal effect relationship](https://www.analyticsvidhya.com/blog/2015/06/establish-causality-events/) between the variables. For example, relationship between rash driving and number of road accidents by a driver is best studied through regression.



Regression methods analysis is an important tool for modelling and analyzing data. Here, we fit a curve / line to the data points, in such a manner that the differences between the distances of data points from the curve or line is minimized.  I’ll explain this in more details in coming sections.

**Regression Models in Machine Learning**

A regression model is a powerful tool in machine learning used for **predicting continuous values** based on the relationship between independent variables (also known as features or predictors) and a dependent variable (also known as target variable).

Here’s a breakdown of how it works:

* **Understanding the Relationship:** Regression analysis helps uncover the connection between the features and the target variable. It essentially explores how the target variable changes when there’s a modification in the features, assuming other factors are constant.
* **Building the Model:** The [machine learning algorithm](https://www.analyticsvidhya.com/blog/2022/01/machine-learning-algorithms/) learns this connection by analyzing a dataset containing examples with known values for both the features and the target variable. This process helps the model identify a function that best represents the observed relationship.
* **Making Predictions:** Once trained, the model can be used to predict the target variable for new, unseen data points. By providing values for the features, the model estimates the corresponding value of the target variable.

**Why Do We Use Regression Analysis?**

As mentioned above, regression analysis estimates the relationship between two or more variables. Let’s understand this with an easy example:

Let’s say, you want to estimate growth in sales of a company based on current economic conditions. You have the recent company data which indicates that the growth in sales is around two and a half times the growth in the economy. Using this insight, we can predict future sales of the company based on current & past information.

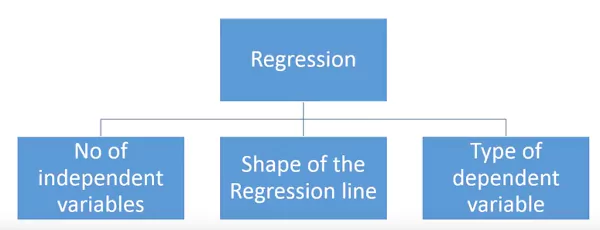
There are multiple benefits of using regression analysis. They are as follows:

* It indicates the **significant relationships** between dependent variable and independent variable.
* It indicates the **strength of impact** of multiple independent variables on a dependent variable.

Regression methods also allows us to compare the effects of variables measured on different scales, such as the effect of price changes and the number of promotional activities. These benefits help market researchers / data analysts / data scientists to eliminate and evaluate the best set of variables to be used for building predictive models.

**Types of Regression Techniques**

There are various kinds of regression techniques available to make predictions. These techniques are mostly driven by three metrics (number of independent variables, type of dependent variables and shape of regression line). We’ll discuss them in detail in the following sections.



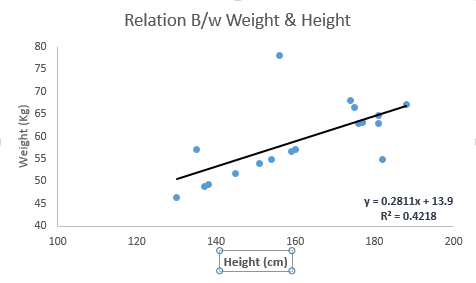
For the creative ones, you can even cook up new regressions, if you feel the need to use a combination of the parameters above, which people haven’t used before. But before you start that, let us understand the most commonly used regressions.

**Linear Regression**

It is one of the most widely known modeling technique. Linear regression is usually among the first few topics which people pick while learning predictive modeling. In this technique, the dependent variable is continuous, independent variable(s) can be [continuous or discrete](https://en.wikipedia.org/wiki/Continuous_and_discrete_variables), and nature of regression line is linear.

Linear Regression establishes a relationship between **dependent variable (Y)** and one or more **independent variables (X)** using a **best fit straight line** (also known as regression line).

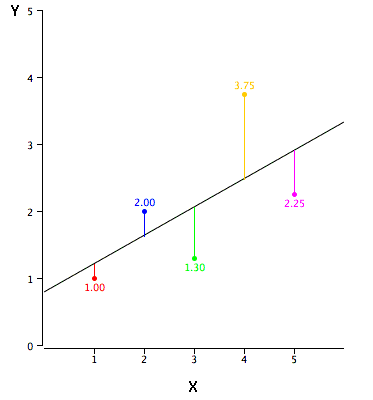
It is represented by an equation **Y=a+b\*X + e**, where a is intercept, b is slope of the line and e is error term. This equation can be used to predict the value of target variable based on given predictor variable(s).



The difference between simple linear regression and multiple linear regression is that, multiple linear regression has (>1) independent variables, whereas simple linear regression has only 1 independent variable.  Now, the question is “How do we obtain best fit line?”.

**How to obtain best fit line (Value of a and b)?**

This task can be easily accomplished by Least Square Method. It is the most common method used for fitting a regression line. It calculates the best-fit line for the observed data by minimizing the sum of the squares of the vertical deviations from each data point to the line. Because the deviations are first squared, when added, there is no cancelling out between positive and negative values.



We can evaluate the model performance using the metric **R-square**. To know more details about these metrics, you can read: Model Performance metrics [Part 1](https://www.analyticsvidhya.com/blog/2015/01/model-performance-metrics-classification/), [Part 2](https://www.analyticsvidhya.com/blog/2015/01/model-perform-part-2/) .

Important Points:

* There must be **linear relationship** between independent and dependent variables
* Multiple regression suffers from **multicollinearity, autocorrelation, heteroskedasticity**.
* Linear Regression is very sensitive to **Outliers**. It can terribly affect the regression line and eventually the forecasted values.
* Multicollinearity can increase the variance of the coefficient estimates and make the estimates very sensitive to minor changes in the model. The result is that the coefficient estimates are unstable
* In case of multiple independent variables, we can go with **forward selection**, **backward elimination** and **step wise approach** for selection of most significant independent variables.

# importing required libraries

import pandas as pd

from sklearn.linear\_model import LinearRegression

from sklearn.metrics import mean\_squared\_error

# reading the train and test dataset

train\_data = pd.read\_csv('train.csv')

test\_data = pd.read\_csv('test.csv')

# shape of the dataset

print('\nShape of training data :',train\_data.shape)

print('\nShape of testing data :',test\_data.shape)

# Now, we need to predict the target variable in the test data

# target variable - Item\_Outlet\_Sales

# seperate the independent and target variable on training data

train\_x = train\_data.drop(columns=['Item\_Outlet\_Sales'],axis=1)

train\_y = train\_data['Item\_Outlet\_Sales']

# seperate the independent and target variable on training data

test\_x = test\_data.drop(columns=['Item\_Outlet\_Sales'],axis=1)

test\_y = test\_data['Item\_Outlet\_Sales']

'''

Create the object of the Linear Regression model

You can also add other parameters and test your code here

Some parameters are : fit\_intercept and normalize

Documentation of sklearn LinearRegression:

https://scikit-learn.org/stable/modules/generated/sklearn.linear\_model.LinearRegression.html

'''

model = LinearRegression()

# fit the model with the training data

model.fit(train\_x,train\_y)

# coefficeints of the trained model

print('\nCoefficient of model :', model.coef\_)

# intercept of the model

print('\nIntercept of model',model.intercept\_)

# predict the target on the test dataset

predict\_train = model.predict(train\_x)

# Root Mean Squared Error on training dataset

rmse\_train = mean\_squared\_error(train\_y,predict\_train)\*\*(0.5)

print('\nRMSE on train dataset : ', rmse\_train)

# predict the target on the testing dataset

predict\_test = model.predict(test\_x)

# Root Mean Squared Error on testing dataset

rmse\_test = mean\_squared\_error(test\_y,predict\_test)\*\*(0.5)

print('\nRMSE on test dataset : ', rmse\_test)

***Also Read:***[***A Beginner’s Guide to Logistic Regression***](https://www.analyticsvidhya.com/blog/2021/08/conceptual-understanding-of-logistic-regression-for-data-science-beginners/)

**Logistic Regression**

Logistic regression is used to find the probability of event=Success and event=Failure. We should use logistic regression when the dependent variable is binary (0/ 1, True/ False, Yes/ No) in nature. Here the value of Y ranges from 0 to 1 and it can represented by following equation.

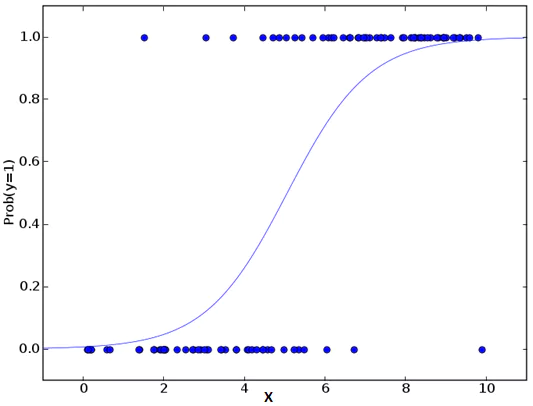
odds= p/ (1-p) = probability of event occurrence / probability of not event occurrence

ln(odds) = ln(p/(1-p))

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

Above, p is the probability of presence of the characteristic of interest. A question that you should ask here is “why have we used log in the equation?”.

Since we are working here with a binomial distribution (dependent variable), we need to choose a link function which is best suited for this distribution. And, it is [**logit**](https://en.wikipedia.org/wiki/Logistic_function) function. In the equation above, the parameters are chosen to maximize the likelihood of observing the sample values rather than minimizing the sum of squared errors (like in ordinary regression).



Also Read: [A Beginner’s Guide to Logistic Regression](https://www.analyticsvidhya.com/blog/2021/08/conceptual-understanding-of-logistic-regression-for-data-science-beginners/)

Important Points:

* Logistic regression is widely used for **classification problems**
* Logistic regression doesn’t require linear relationship between dependent and independent variables.  It can handle various types of relationships because it applies a non-linear log transformation to the predicted odds ratio
* To avoid over fitting and under fitting, we should include all significant variables. A good approach to ensure this practice is to use a step wise method to estimate the logistic regression methods.
* It requires **large sample sizes** because maximum likelihood estimates are less powerful at low sample sizes than ordinary least square
* The independent variables should not be correlated with each other i.e. **no multi collinearity**.  However, we have the options to include interaction effects of categorical variables in the analysis and in the model.
* If the values of dependent variable is ordinal, then it is called as **Ordinal logistic regression**
* If dependent variable is multi class then it is known as **Multinomial Logistic regression**.

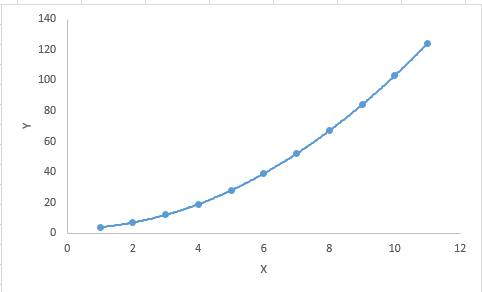
***Note: You can understand the above regression techniques in a video format –***[***Fundamentals of Regression Analysis***](https://courses.analyticsvidhya.com/courses/Fundamentals-of-Regression-Analysis?utm_source=blog&utm_medium=introduction_to_regression)

**Polynomial Regression**

A regression equation is a polynomial regression equation if the power of independent variable is more than 1. The equation below represents a polynomial equation:

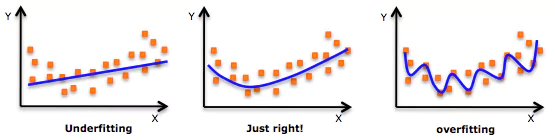
y=a+b\*x^2

In this regression technique, the best fit line is not a straight line. It is rather a curve that fits into the data points.



Important Points:

* While there might be a temptation to fit a higher degree polynomial to get lower error, this can result in over-fitting. Always plot the relationships to see the fit and focus on making sure that the curve fits the nature of the problem. Here is an example of how plotting can help:



* Especially look out for curve towards the ends and see whether those shapes and trends make sense. Higher polynomials can end up producing wierd results on extrapolation.

***Also Read:***[***Understanding Polynomial Regression Model***](https://www.analyticsvidhya.com/blog/2021/10/understanding-polynomial-regression-model/)

**Stepwise Regression**

Researchers use this form of regression when dealing with multiple independent variables. In this technique, an automatic process selects the independent variables, with no human intervention.

Researchers achieve this feat by observing statistical values like R-square, t-stats, and AIC metric to discern significant variables. They fit the regression model in Stepwise regression by adding or dropping covariates one at a time based on a specified criterion.

* Standard stepwise regression does two things. It adds and removes predictors as needed for each step.
* Forward selection starts with most significant predictor in the model and adds variable for each step.
* Backward elimination starts with all predictors in the model and removes the least significant variable for each step.

The aim of this modeling technique is to maximize the prediction power with minimum number of predictor variables. It is one of the method to handle[higher dimensionality](https://www.analyticsvidhya.com/blog/2015/07/dimension-reduction-methods/) of data set.

**Ridge Regression**

Ridge Regression is a technique used when the data suffers from multicollinearity (independent variables are highly correlated). In multicollinearity, even though the least squares estimates (OLS) are unbiased, their variances are large which deviates the observed value far from the true value. By adding a degree of bias to the regression estimates, ridge regression reduces the standard errors.

Above, we saw the equation for linear regression. Remember? It can be represented as:

y=a+ b\*x

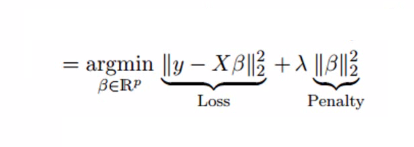
This equation also has an error term. The complete equation becomes:

y=a+b\*x+e (error term),  [error term is the value needed to correct for a prediction error between the observed and predicted value]

=> y=a+y= a+ b1x1+ b2x2+....+e, for multiple independent variables.

In a linear equation, prediction errors can be decomposed into two sub components. First is due to the **biased** and second is due to the **variance**. Prediction error can occur due to any one of these two or both components. Here, we’ll discuss about the error caused due to variance.

Ridge regression solves the multicollinearity problem through [shrinkage parameter](https://en.wikipedia.org/wiki/Shrinkage_estimator) λ (lambda). Look at the equation below.

[](https://www.analyticsvidhya.com/wp-content/uploads/2015/08/Ridge2.png)

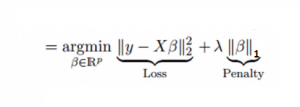
In this equation, we have two components. First one is least square term and other one is lambda of the summation of β2 (beta- square) where β is the coefficient. This is added to least square term in order to shrink the parameter to have a very low variance.

Important Points:

* The assumptions of this regression is same as least squared regression except normality is not to be assumed
* Ridge regression shrinks the value of coefficients but doesn’t reaches zero, which suggests no feature selection feature
* This is a regularization method and uses [l2 regularization](https://en.wikipedia.org/wiki/Regularization_(mathematics)).

***Also Read:***[***Ridge and Lasso Regression in Python***](https://www.analyticsvidhya.com/blog/2016/01/ridge-lasso-regression-python-complete-tutorial/)

**Lasso Regression**

[](https://www.analyticsvidhya.com/wp-content/uploads/2015/08/Lasso.png)

Similar to Ridge Regression, Lasso (Least Absolute Shrinkage and Selection Operator) also penalizes the absolute size of the regression coefficients. In addition, it is capable of reducing the variability and improving the accuracy of linear regression models.  Look at the equation below: Lasso regression differs from ridge regression in a way that it uses absolute values in the penalty function, instead of squares. This leads to penalizing (or equivalently constraining the sum of the absolute values of the estimates) values which causes some of the parameter estimates to turn out exactly zero. Larger the penalty applied, further the estimates get shrunk towards absolute zero. This results to variable selection out of given n variables.

Important Points:

* The assumptions of lasso regression is same as least squared regression except normality is not to be assumed
* Lasso Regression shrinks coefficients to zero (exactly zero), which certainly helps in feature selection
* Lasso is a regularization method and uses [l1 regularization](https://en.wikipedia.org/wiki/Regularization_(mathematics))
* If group of predictors are highly correlated, lasso picks only one of them and shrinks the others to zero

**ElasticNet Regression**

ElasticNet is hybrid of Lasso and Ridge Regression techniques. It is trained with L1 and L2 prior as regularizer. Elastic-net is useful when there are multiple features which are correlated. Lasso is likely to pick one of these at random, while elastic-net is likely to pick both.

elastic net regression

A practical advantage of trading-off between Lasso and Ridge is that, it allows Elastic-Net to inherit some of Ridge’s stability under rotation.

Important Points:

* It encourages group effect in case of highly correlated variables
* There are no limitations on the number of selected variables
* It can suffer with double shrinkage

Beyond these 7 most commonly used regression techniques, you can also look at other models like [Bayesian](https://en.wikipedia.org/wiki/Bayesian_linear_regression), [Ecological](https://en.wikipedia.org/wiki/Ecological_regression) and [Robust regression](https://en.wikipedia.org/wiki/Robust_regression).

**How to Select the Right Regression Model?**

Life is usually simple, when you know only one or two techniques. One of the training institutes I know of tells their students – if the outcome is continuous – apply linear regression. If it is binary – use logistic regression! However, higher the number of options available at our disposal, more difficult it becomes to choose the right one. A similar case happens with [regression models](https://www.analyticsvidhya.com/blog/2022/01/different-types-of-regression-models/).

Within multiple types of regression models, it is important to choose the best suited technique based on type of independent and dependent variables, dimensionality in the data and other essential characteristics of the data.

**Factors to Consider While Selecting Regression Model**

Below are the key factors that you should practice to select the right regression model:

* Data exploration is an inevitable part of building predictive model. It should be you first step before selecting the right model like identify the relationship and impact of variables
* To compare the goodness of fit for different models, we can analyse different metrics like statistical significance of parameters, R-square, Adjusted r-square, AIC, BIC and error term. Another one is the [Mallow’s Cp](http://support.minitab.com/en-us/minitab/17/topic-library/modeling-statistics/regression-and-correlation/goodness-of-fit-statistics/what-is-mallows-cp/) criterion. This essentially checks for possible bias in your model, by comparing the model with all possible submodels (or a careful selection of them).
* Cross-validation is the best way to evaluate models used for prediction. Here you divide your data set into two group (train and validate). A simple mean squared difference between the observed and predicted values give you a measure for the prediction accuracy.
* If your data set has multiple confounding variables, you should not choose automatic model selection method because you do not want to put these in a model at the same time.
* It’ll also depend on your objective. It can occur that a less powerful model is easy to implement as compared to a highly statistically significant model.
* Regression regularization methods(Lasso, Ridge and ElasticNet) works well in case of high dimensionality and multicollinearity among the variables in the data set.

**Techniques to Estimate Parameters in Linear Regression**

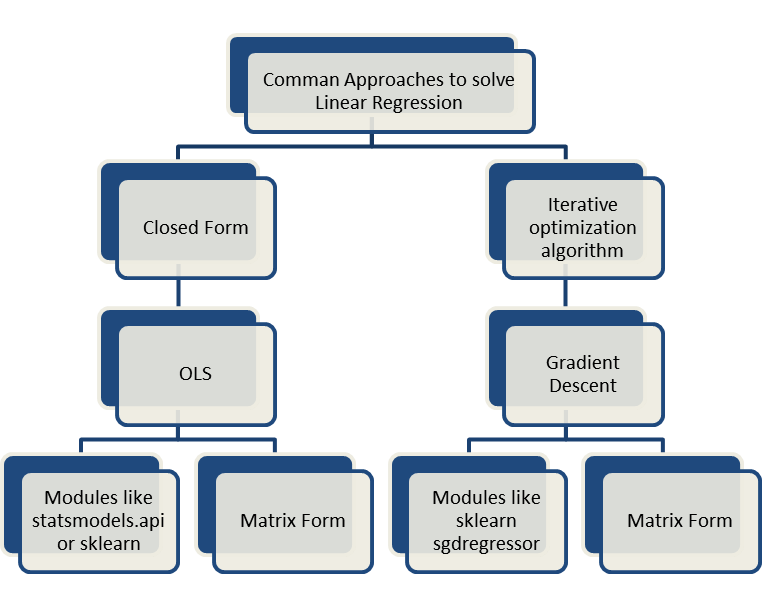
Here are the **main fundamentally different estimation techniques** for linear regression parameters:

| **#** | **Technique** | **Core Idea** | **Is it Fundamentally Different from OLS?** |
| --- | --- | --- | --- |
| 1 | **OLS (Ordinary Least Squares)** | Minimize sum of squared errors | 🔹 Base method |
| 2 | **Gradient Descent (GD)** | Iteratively minimize the same OLS loss | ✅ Different optimization strategy |
| 3 | **Maximum Likelihood Estimation (MLE)** | Maximize the likelihood under normal error assumption | ✅ Based on probability theory |
| 4 | **Bayesian Estimation** | Compute a posterior distribution using prior and likelihood | ✅ Fundamentally different philosophy |
| 5 | **Generalized Least Squares (GLS)** | Adjust for correlated and non-constant variance errors | ✅ Different covariance assumptions |
| 6 | **Weighted Least Squares (WLS)** | Assign weights to data points (for heteroscedasticity) | ✅ Different error modeling |
| 7 | **Robust Regression** (M-estimators, Huber Loss) | Minimize robust loss less sensitive to outliers | ✅ Uses different loss function, not squared error |
| 8 | **Theil–Sen Estimator** | Uses **medians** of slopes, not squared errors | ✅ Non-parametric, median-based |
| 9 | **Quantile Regression** | Minimize quantile loss (pinball loss) instead of squared error | ✅ Different objective — quantiles not means |

These are **penalized OLS methods** — they modify the same OLS loss function but don't change the estimation framework itself:

| **Method** | **Reason it's not fundamentally different** |
| --- | --- |
| Ridge Regression (L2) | OLS + L2 penalty |
| Lasso Regression (L1) | OLS + L1 penalty |
| ElasticNet | OLS + L1 + L2 |
| Tikhonov Regularization | Variant of Ridge |
| Early stopping | Just a training-time technique on gradient descent |

| **Technique** | **AI Context** | **Why It's Used** | **Notes** |
| --- | --- | --- | --- |
| **Gradient Descent (and variants)** | 🔥 Universal | Scalable, general-purpose | Most deep learning models use it |
| **Stochastic Gradient Descent (SGD)** | Neural nets, online learning | Efficient with large data | Mini-batch SGD is the standard |
| **Maximum Likelihood Estimation (MLE)** | Probabilistic models, classification | Strong statistical foundation | Logistic regression, Naive Bayes |
| **Regularized Regression (Ridge/Lasso/ElasticNet)** | Feature selection, linear models | Controls overfitting, improves generalization | Very common in ML pipelines |
| **Bayesian Estimation** | Probabilistic programming, uncertainty modeling | Produces posterior distributions | Used in Bayesian ML, but more niche |
| **Robust Regression** | ML pipelines with noisy data | Handles outliers better | Less common in deep learning, more in classical ML |
| **Quantile Regression** | Time series, risk modeling | Predict specific quantiles, not just means | Useful in finance and forecasting |
| **Reinforcement Learning Techniques** | Agent models | Often uses stochastic estimators (like policy gradients) | Outside classic regression scope but AI core |
| **OLS (Ordinary Least Squares)** | Teaching, simple ML models | Easy to explain and implement | Rare in large-scale AI systems |
| **Matrix Decomposition (SVD, QR)** | Dimensionality reduction | Used in PCA, recommender systems | More in preprocessing than model training |



**Ordinary Least Squares (OLS):**

<https://www.analyticsvidhya.com/blog/2023/01/a-comprehensive-guide-to-ols-regression-part-1/>

**What is the OLS minimization problem?**

At the core of OLS regression lies an optimization challenge: finding the line (or hyperplane in higher dimensions) that best fits the data. But what does "best fit" mean? "Best fit" here means minimizing the sum of squared residuals.

Let me try to explain the minimizing problem while also explaining the idea of residuals.

* **Residuals Explained:** Residuals are the differences between the actual observed values and the values predicted by the regression model. For each data point, the residual tells us how far off our prediction was.
* **Why Square the Residuals?** By squaring each residual, we ensure that positive and negative differences don't cancel each other out. Squaring also gives more weight to larger errors, meaning the model prioritizes reducing bigger mistakes.

By minimizing the sum of the squared residuals, the regression line become an accurate representation of the relationship between the independent and dependent variables. In fact, by minimizing the sum of squared residuals, our model has the smallest possible overall error in its predictions. To learn more about residuals and regression decomposition, read our tutorial, [**Understanding Sum of Squares: A Guide to SST, SSR, and SSE**](https://www.datacamp.com/tutorial/regression-sum-of-squares).

**What is the ordinary least squares estimator?**

In the context of regression, estimators are used to calculate the coefficients that describe the relationship between independent variables and the dependent variable. The ordinary least squares (OLS) estimator is one such method. It finds the coefficient values that minimize the sum of the squared differences between the observed values and those predicted by the model.

I'm bringing this up to keep the terms clear. Regression could be done with other estimators, each offering different advantages depending on the data and the analysis goals. For instance, some estimators are more robust to outliers, while others help prevent overfitting by regularizing the model parameters.

**How are the OLS regression parameters estimated?**

To determine the coefficients that best fit the regression model, the OLS estimator employs mathematical techniques to minimize the sum of squared residuals. One possible method is [**the normal equation**](https://www.datacamp.com/tutorial/tutorial-normal-equation-for-linear-regression), which provides a direct solution by setting up a system of equations based on the data and solving for the coefficients that achieve the smallest possible sum of squared differences between the observed and predicted values.

However, solving the normal equation can become computationally demanding, especially with large datasets. To address this, another technique called [**QR decomposition**](https://www.datacamp.com/tutorial/qr-decomposition) is often used. QR decomposition breaks down the matrix of independent variables into two simpler matrices: an orthogonal matrix (Q) and an upper triangular matrix (R). This simplification makes the calculations more efficient and it also improves numerical stability.

**Ordinary Least Squares Estimation**

The goal of OLS is to find the coefficient vector 𝛽 that minimizes the sum of squared residuals (SSR). The residual for each observation is the difference between the observed value and the predicted value:

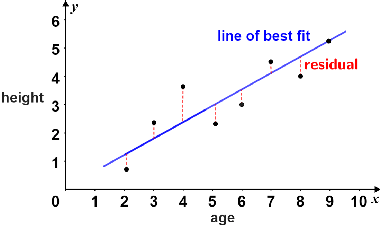
OLS regression works its magic by finding the perfect coefficients that create a line — let’s call it the ‘line of best fit’ — which represents the relationship between study hours and scores. This line is like a conductor’s baton, directing the melody of data points in the symphony of statistics.

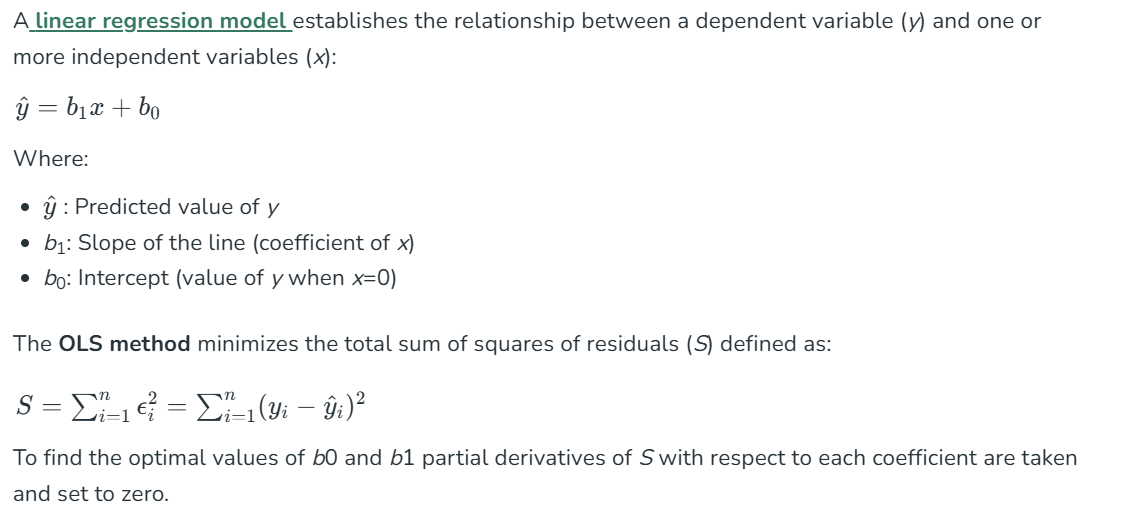
The equation of this line is



where *Y* is the predicted test score, *β*0​ is the intercept (the starting score, if you will), and *β*1​ is the slope (how much the score changes with each additional hour of study).







* The slope *β*1​ is given by the formula:



* The intercept *β*0​ is calculated using the mean values of X and Y:



* With these steps, we arrive at the coefficients that define the line of best fit according to the OLS criterion.

**When to Use OLS Regression**

How do we decide to use OLS regression? In making that decision, we have to both assess the characteristics of our dataset and we also have to define the specific problem we are trying to solve.

**Assumptions of OLS regression**

Before applying OLS regression, we should make sure that our data meets the following assumptions so that we have reliable results:

1. **Linearity**: The relationship between independent and dependent variables must be linear.
2. **Independence of errors**: Residuals should be uncorrelated with each other.
3. **Homoscedasticity**: Residuals should have constant variance across all levels of the independent variables.
4. **Normality of errors**: Residuals should be normally distributed.

Serious violations of these assumptions can lead to biased estimates or unreliable predictions. Therefore, we really hav to assess and address any potential issues before going further.

**Applications of OLS regression**

Once the assumptions are satisfied, OLS regression can be used for different purposes:

* **Predictive modeling**: Forecasting outcomes such as sales, revenue, or trends.
* **Relationship analysis**: Understanding the influence of independent variables on a dependent variable.
* **Hypothesis testing**: Assessing whether specific predictors significantly impact the outcome variable.

**Deeper Insights into OLS Regression**

Now that we explored the basics of OLS regression, let's explore some more advanced concepts.

**OLS regression and maximum likelihood estimation**

Maximum likelihood estimation (MLE) is another concept talked about alongside OLS regression, and for good reason. We have spent time so far talking about how OLS minimizes the sum of squared residuals to estimate coefficients. Let’s now take a step back to talk about MLE.

MLE maximizes the likelihood of observing the given data under our model. It works by assuming a specific probability distribution for the error term. This probability distribution is usually a [**normal, or Gaussian, distribution**](https://www.datacamp.com/tutorial/gaussian-distribution). Using our probability distribution, we find parameter values that make the observed data most probable.

The reason I’m bringing up maximum likelihood estimation right now is because, in the context of OLS regression, the MLE approach leads to the same coefficient estimates as we get by minimizing the sum of squares errors, provided that the errors are normally distributed.

**Interpreting OLS regression as a weighted average**

Another fascinating perspective on OLS regression is its interpretation as a weighted average. Prof. Andrew Gelman discusses the idea that the coefficients in an OLS regression can be thought of as a weighted average of the observed data points, where the weights are determined by the variance of the predictors and the structure of the model.

This view provides some insight into how the regression process works and why it behaves the way it does because OLS regression is really giving more weight to observations that have less variance or are closer to the model's predictions. You can also tune into our DataFramed podcast episode, [**Election Forecasting and Polling**](https://www.datacamp.com/podcast/election-forecasting-and-polling), to hear what Professor Gelman says about using regression in election polling.

**OLS Regression vs. Similar Regression Methods**

Several other regression methods have names that might sound similar but serve different purposes or operate under different assumptions. Let's take a look at some similar-sounding ones:

**OLS vs. weighted least squares (WLS)**

WLS is an extension of OLS that assigns different weights to each data point based on the variance of their observations. WLS is particularly useful when the assumption of constant variance of residuals is violated. By weighting observations inversely to their variance, WLS provides more reliable estimates when dealing with [**heteroscedastic data**](https://www.datacamp.com/tutorial/heteroscedasticity).

**OLS vs. partial least squares (PLS) regression**

PLS combines features of [**principal component analysis**](https://www.datacamp.com/tutorial/pca-analysis-r) and [**multiple regression**](https://www.datacamp.com/tutorial/multiple-linear-regression-r-tutorial) by extracting latent variables that capture the maximum covariance between predictors and the response variable. PLS is advantageous in situations with [**multicollinearity**](https://www.datacamp.com/tutorial/multicollinearity) or when the number of predictors exceeds the number of observations. It reduces dimensionality while simultaneously maximizing the predictive power, which OLS does not inherently address.

**OLS vs. generalized least squares (GLS)**

Similar to WLS, GLS generalizes OLS by allowing for correlated and/or non-constant variance of the residuals. GLS adjusts the estimation process to account for violations of OLS assumptions regarding the residuals, providing more efficient and unbiased estimates in such scenarios.

**OLS vs. total least squares (TLS)**

Also known as orthogonal regression, TLS minimizes the perpendicular distances from the data points to the regression line, rather than the vertical distances minimized by OLS. TLS is useful when there is error in both the independent and dependent variables, whereas OLS assumes that only the dependent variable has measurement error.

**Alternatives to OLS Regression**

When the relationship between variables is complex or nonlinear, **non-parametric** regression methods offer flexible alternatives to OLS by allowing the data to determine the form of the regression function. All of the previous examples (the "similar-sounding" ones) belong to the category of parametric models. But non-parametric models could also be used when you want to model patterns without the constraints of parametric assumptions.

| **Method** | **Description** | **Advantages** | **Common Use Cases** |
| --- | --- | --- | --- |
| Kernel Regression | Uses weighted averages with a kernel to smooth data. | Captures nonlinear relationships Flexible smoothing | Exploratory analysis Unknown variable relationships |
| Local Regression | Fits local polynomials to subsets of data for a smooth curve. | Handles complex patterns Adaptive smoothness | Trend visualization Scatterplot smoothing |
| Regression Trees | Splits data into branches to fit simple models in each segment. | Easy to interpret Handles interactions | Segmenting data Identifying distinct data regimes |
| Spline Regression | Uses piecewise polynomials with continuity at knots to model data. | Models smooth nonlinear trends Flexible fitting | Time series Growth curves |