**Regularization in machine learning means ‘simplifying the outcome’. It is a technique used to prevent overfitting and improve the performance of models. In case a model is overfitting and too complex, you can use regularization to make the model generalize better. You should use regularization if the gap in performance between train and test is big. This means the model grasps too much details of the train set. Overfitting is related to high variance, which means the model is sensitive to specific samples of the train set.**

It means the model is not able to predict the output or target column for the unseen data by introducing noise in the output, and hence the model is called an **overfitted model**.

**By noise we mean those data points in the dataset which don’t really represent the true properties of your data, but only due to a random chance.**

So, to deal with the problem of overfitting we take the help of regularization techniques.

**What is Regularization?**

* 👉 It is one of the most important concepts of machine learning. This technique prevents the model from overfitting by adding **extra information** to it.
* 👉 It is a form of regression that shrinks the coefficient estimates towards zero. In other words, this technique forces us not to learn a more            complex or flexible model, to avoid the problem of overfitting.
* 👉 Now, let’s understand the**“How flexibility of a model is represented?”**
* For regression problems, **the increase in flexibility of a model is represented by an increase in its coefficients**, which are calculated              from the regression line.
* 👉 In simple words, **“In the Regularization technique, we reduce the magnitude of the independent variables by keeping the same                number of variables”.**It maintains accuracy as well as a generalization of the model.

**How does Regularization Work?**

Regularization works by adding a penalty or complexity term or shrinkage term with Residual Sum of Squares (RSS) to the complex model.

Let’s consider the **Simple linear regression**equation:

Here Y represents the dependent feature or response which is the learned relation. Then,

Y is approximated to**β0 + β1X1 + β2X2 + …+ βpXp**

Here, **X1, X2, …Xp** are the independent features or predictors for Y, and

**β0, β1,…..βn** represents the coefficients estimates for different variables or predictors(X), which describes the weights or magnitude attached to the features, respectively.

In simple linear regression, our optimization function or loss function is known as the **residual sum of squares (RSS).**

We choose those set of coefficients, such that the following loss function is minimized:

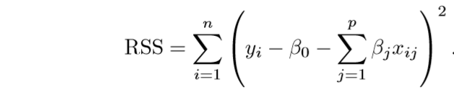


Fig. Cost Function For Simple Linear Regression

Now, this will adjust the coefficient estimates based on the training data. If there is noise present in the training data, then the estimated coefficients won’t generalize well and are not able to predict the future data.

This is where regularization comes into the picture, which shrinks or regularizes these learned estimates towards zero, by adding a loss function with optimizing parameters to make a model that can predict the accurate value of Y.

**Role Of Regularization**

In Python, Regularization is a technique used to prevent overfitting by adding a penalty term to the loss function, discouraging the model from assigning too much importance to individual features or coefficients.  
Let’s explore some more detailed explanations about the role of Regularization in Python:

1. **Complexity Control**: Regularization helps control model complexity by preventing overfitting to training data, resulting in better generalization to new data.
2. **Preventing Overfitting**: One way to prevent overfitting is to use regularization, which penalizes large coefficients and constrains their magnitudes, thereby preventing a model from becoming overly complex and memorizing the training data instead of learning its underlying patterns.
3. **Balancing Bias and Variance**: Regularization can help balance the trade-off between model bias (underfitting) and model variance (overfitting) in machine learning, which leads to improved performance.
4. **Feature Selection**: Some regularization methods, such as L1 regularization (Lasso), promote sparse solutions that drive some feature coefficients to zero. This automatically selects important features while excluding less important ones.
5. **Handling Multicollinearity**: When features are highly correlated (multicollinearity), regularization can stabilize the model by reducing coefficient sensitivity to small data changes.
6. **Generalization**: Regularized models learn underlying patterns of data for better generalization to new data, instead of memorizing specific examples.

**Regularization in Machine Learning**

Regularization is a technique used to reduce errors by fitting the function appropriately on the given training set and avoiding overfitting. The commonly used [regularization techniques](https://www.geeksforgeeks.org/lasso-vs-ridge-vs-elastic-net-ml/) are :

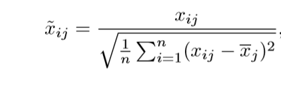
1. Lasso Regularization – L1 Regularization
2. Ridge Regularization – L2 Regularization
3. Elastic Net Regularization

**Lasso Regression**

A regression model which uses the **L1 Regularization**technique is called **LASSO(Least Absolute Shrinkage and Selection Operator)** regression. **Lasso Regression** adds the *“absolute value of magnitude”* of the coefficient as a penalty term to the loss function(L). Lasso regression also helps us achieve feature selection by penalizing the weights to approximately equal to zero if that feature does not serve any purpose in the model.

Cost=1n∑i=1n(yi−yi^)2+λ∑i=1m∣wi∣Cost=n1​∑i=1n​(yi​−yi​^​)2+*λ*∑i=1m​∣wi​∣

*where,*

* ***m****– Number of Features*
* ***n****– Number of Examples*
* ***y\_i****– Actual Target Value*
* ***y\_i(hat)****– Predicted Target Value*
* ***RSS is modified by adding the shrinkage quantity.*** Now, the coefficients are estimated by minimizing this function. Here, ***λ is the tuning parameter that decides how much we want to penalize the flexibility of our model.*** The increase in flexibility of a model is represented by increase in its coefficients, and if we want to minimize the above function, then these coefficients need to be small. This is how the Ridge regression technique prevents coefficients from rising too high. Also, notice that we shrink the estimated association of each variable with the response, except the intercept β0, This intercept is a measure of the mean value of the response when xi1 = xi2 = …= xip = 0.
* *When λ = 0, the penalty term has no eﬀect*, and the estimates produced by ridge regression will be equal to least squares. However, ***as λ→∞, the impact of the shrinkage penalty grows, and the ridge regression coeﬃcient estimates will approach zero***. As can be seen, selecting a good value of λ is critical. Cross validation comes in handy for this purpose. The coefficient estimates produced by this method are ***also known as the L2 norm***.
* ***The coefficients that are produced by the standard least squares method are scale equivariant***, i.e. if we multiply each input by c then the corresponding coefficients are scaled by a factor of 1/c. Therefore, regardless of how the predictor is scaled, the multiplication of predictor and coefficient(Xjβj) remains the same. ***However, this is not the case with ridge regression, and therefore, we need to standardize the predictors or bring the predictors to the same scale before performing ridge regression***. The formula used to do this is given below.
* 

**Ridge Regression**

A regression model that uses the **L2 regularization** technique is called **Ridge regression**. **Ridge regression** adds the “*squared magnitude*” of the coefficient as a penalty term to the loss function(L).

Cost=1n∑i=1n(yi−yi^)2+λ∑i=1mwi2Cost=n1​∑i=1n​(yi​−yi​^​)2+*λ*∑i=1m​wi2​

**Elastic Net Regression**

This model is a combination of L1 as well as L2 regularization. That implies that we add the absolute norm of the weights as well as the squared measure of the weights. With the help of an extra [hyperparameter](https://www.geeksforgeeks.org/hyperparameter-tuning/) that controls the ratio of the L1 and L2 regularization.

Cost=1n∑i=1n(yi−yi^)2+λ((1−α)∑i=1m∣wi∣+α∑i=1mwi2)Cost=n1​∑i=1n​(yi​−yi​^​)2+*λ*((1−*α*)∑i=1m​∣wi​∣+*α*∑i=1m​wi2​)

**Benefits of Regularization**

1. Regularization improves model generalization by reducing overfitting. Regularized models learn underlying patterns, while overfit models memorize noise in training data.
2. Regularization techniques such as L1 (Lasso) L1 regularization simplifies models and improves interpretability by reducing coefficients of less important features to zero.
3. Regularization improves model performance by preventing excessive weighting of outliers or irrelevant features.
4. Regularization makes models stable across different subsets of the data. It reduces the sensitivity of model outputs to minor changes in the training set.
5. Regularization prevents models from becoming overly complex, which is especially important when dealing with limited data or noisy environments.
6. Regularization can help handle multicollinearity (high correlation between features) by reducing the magnitudes of correlated coefficients.
7. Regularization introduces hyperparameters (e.g., alpha or lambda) that control the strength of regularization. This allows fine-tuning models to achieve the right balance between bias and variance.
8. Regularization promotes consistent model performance across different datasets. It reduces the risk of dramatic performance changes when encountering new data.

Regularization may be defined as any modification or change in the learning[algorithm](https://www.analyticsvidhya.com/blog/2022/08/regularization-in-machine-learning/) that helps reduce its error over a test dataset, commonly known as generalization error but not on the supplied or training dataset.

In learning algorithms, there are many variants of regularization techniques, each of which tries to cater to different challenges. These can be listed down straightforwardly based on the kind of challenge the technique is trying to deal with:

1. Some try to put extra constraints on the learning of an ML model, like adding restrictions on the range/type of parameter values.
2. Some add more terms in the objective or cost function, like a soft constraint on the parameter values. More often than not, a careful selection of the right constraints and penalties in the cost function contributes to a massive boost in the model's performance, specifically on the test dataset.
3. These extra terms can also be encoded based on some prior information that closely relates to the dataset or the problem statement.
4. One of the most commonly used regularization techniques is creating ensemble models, which take into account the collective decision of multiple models, each trained with different samples of data.

The main aim of regularization is to reduce the over-complexity of the machine learning models and help the model learn a simpler function to promote generalization.

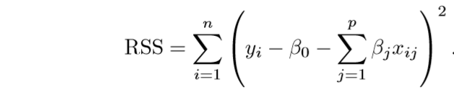
**Regularization**

This is a form of regression, that constrains/ regularizes or shrinks the coefficient estimates towards zero. In other words, ***this technique discourages learning a more complex or flexible model, so as to avoid the risk of overfitting.***

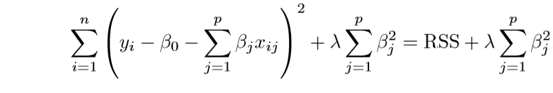
A simple relation for linear regression looks like this. Here Y represents the learned relation and *β represents the coefficient estimates for different variables or predictors(X).*

***Y ≈ β0 + β1X1 + β2X2 + …+ βpXp***

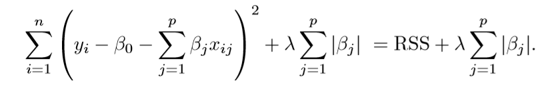
The fitting procedure involves a loss function, known as residual sum of squares or RSS. The coefficients are chosen, such that they minimize this loss function.



Now, this will adjust the coefficients based on your training data. *If there is noise in the training data, then the estimated coefficients won’t generalize well to the future data. This is where regularization comes in and shrinks or regularizes these learned estimates towards zero.*



**Lasso**



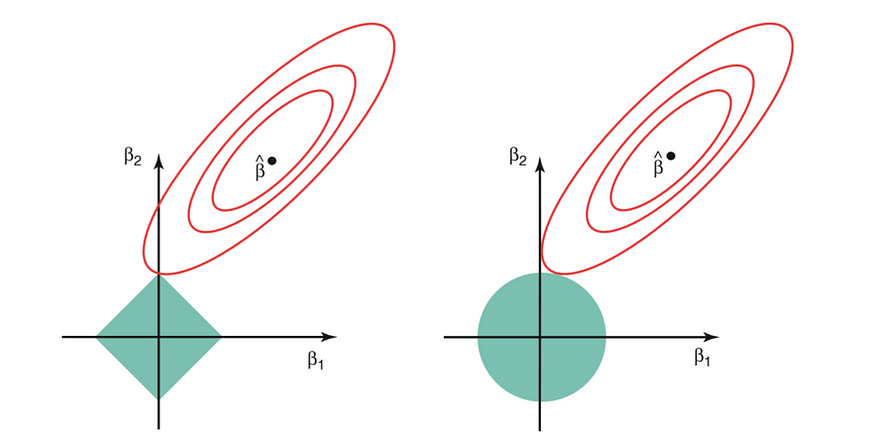
Lasso is another variation, in which the above function is minimized. Its clear that ***this variation differs from ridge regression only in penalizing the high coefficients***. It uses |βj|(modulus)instead of squares of β, as its penalty. In statistics, this is***known as the L1 norm***.

Lets take a look at above methods with a different perspective. *The ridge regression can be thought of as solving an equation, where summation of squares of coefficients is less than or equal to s*. And *the Lasso can be thought of as an equation where summation of modulus of coefficients is less than or equal to s*. Here, s is a constant that exists for each value of shrinkage factor *λ.****These equations are also referred to as constraint functions.***

***Consider their are 2 parameters in a given problem***. Then according to above formulation, the ***ridge regression is expressed by β1² + β2² ≤ s***. This implies that *ridge regression coefficients have the smallest RSS(loss function) for all points that lie within the circle given by β1² + β2² ≤ s.*

Similarly, ***for lasso, the equation becomes,|β1|+|β2|≤ s***. This implies that *lasso coefficients have the smallest RSS(loss function) for all points that lie within the diamond given by |β1|+|β2|≤ s.*

The image below describes these equations.



Credit : An Introduction to Statistical Learning by Gareth James, Daniela Witten, Trevor Hastie, Robert Tibshirani

***The above image shows the constraint functions(green areas), for lasso(left) and ridge regression(right), along with contours for RSS(red ellipse)***. Points on the ellipse share the value of RSS. For a very large value of s, the green regions will contain the center of the ellipse, making coefficient estimates of both regression techniques, equal to the least squares estimates. But, this is not the case in the above image. In this case, the lasso and ridge regression coefficient estimates are given by the ﬁrst point at which an ellipse contacts the constraint region. ***Since ridge regression has a circular constraint with no sharp points, this intersection will not generally occur on an axis, and so the ridge regression coeﬃcient estimates will be exclusively non-zero.*** ***However, the lasso constraint has corners at each of the axes, and so the ellipse will often intersect the constraint region at an axis. When this occurs, one of the coeﬃcients will equal zero.*** In higher dimensions(where parameters are much more than 2), many of the coeﬃcient estimates may equal zero simultaneously.

**This sheds light on the obvious disadvantage of ridge regression, which is model interpretability.** It will shrink the coefficients for least important predictors, very close to zero. But it will never make them exactly zero. In other words, the final model will include all predictors. However, in the case of the lasso, the L1 penalty has the eﬀect of forcing some of the coeﬃcient estimates to be exactly equal to zero when the tuning parameter λ is suﬃciently large. **Therefore, the lasso method also performs variable selection and is said to yield sparse models.**

**What does Regularization achieve?**

A standard least squares model tends to have some variance in it, i.e. this model won’t generalize well for a data set different than its training data. ***Regularization, significantly reduces the variance of the model, without substantial increase in its bias***. So the tuning parameter λ, used in the regularization techniques described above, controls the impact on bias and variance. As the value of λ rises, it reduces the value of coefficients and thus reducing the variance. ***Till a point, this increase in λ is beneficial as it is only reducing the variance(hence avoiding overfitting), without loosing any important properties in the data.*** But after certain value, the model starts loosing important properties, giving rise to bias in the model and thus underfitting. Therefore, the value of λ should be carefully selected.

This is all the basic you will need, to get started with Regularization. It is a useful technique that can help in improving the accuracy of your regression models. A popular library for implementing these algorithms is [**Scikit-Learn**](https://becominghuman.ai/implementing-decision-trees-using-scikit-learn-5057b27221ec). It has a wonderful api that can get your model up an running with **just a few lines of code in python**.

[Regularization in Machine Learning | Analytics Vidhya](https://www.analyticsvidhya.com/blog/2022/08/regularization-in-machine-learning/)

[Regularization: A Method to Solve Overfitting in Machine Learning | by Bassant Gamal | Analytics Vidhya | Medium](https://medium.com/analytics-vidhya/regularization-a-method-to-solve-overfitting-in-machine-learning-ed5f13647b91)

[Regularization in Machine Learning (with Code Examples) (dataquest.io)](https://www.dataquest.io/blog/regularization-in-machine-learning/)

[Regularization in Machine Learning | by Ritwick Roy | Towards Data Science](https://towardsdatascience.com/regularization-in-machine-learning-6fbc4417b1e5)

[Everything You Need To Know About Regularization | by Hennie de Harder | Towards Data Science](https://towardsdatascience.com/everything-you-need-to-know-about-regularization-64734f240622)