There are three error metrics that are commonly used for evaluating and reporting the performance of a regression model; they are:

- Mean Squared Error (MSE).
- Root Mean Squared Error (RMSE).
- Mean Absolute Error (MAE)

# Metrics for Regression

In this section, we will take a closer look at the popular metrics for regression models and how to calculate them for your predictive modelling project.

## Mean Squared Error

Mean Squared Error, or MSE for short, is a popular error metric for regression problems.

It is also an important loss function for algorithms fit or optimized using the least squares framing of a regression problem. Here "*least squares*" refers to minimizing the mean squared error between predictions and expected values.

The MSE is calculated as the mean or average of the squared differences between predicted and expected target values in a dataset.

- MSE = 1 / N * sum for i to N (y_i – yhat_i)^2

Where *y_i* is the i'th expected value in the dataset and *yhat_i* is the i'th predicted value. The difference between these two values is squared, which has the effect of removing the sign, resulting in a positive error value.

The squaring also has the effect of inflating or magnifying large errors. That is, the larger the difference between the predicted and expected values, the larger the resulting squared positive error. This has the effect of "*punishing*" models more for larger errors when MSE is used as a loss function. It also has the effect of "*punishing*" models by inflating the average error score when used as a metric.

A perfect mean squared error value is 0.0, which means that all predictions matched the expected values exactly.

This is almost never the case, and if it happens, it suggests your predictive modelling problem is trivial.

A good MSE is relative to your specific dataset.

It is a good idea to first establish a baseline MSE for your dataset using a naive predictive model, such as predicting the mean target value from the training dataset. A model that achieves an MSE better than the MSE for the naive model has skill.

## Root Mean Squared Error

The Root Mean Squared Error, or RMSE, is an extension of the mean squared error.
Importantly, the square root of the error is calculated, which means that the units of the RMSE are the same as the original units of the target value that is being predicted.

For example, if your target variable has the units "*dollars*," then the RMSE error score will also have the unit "*dollars*" and not "*squared dollars*" like the MSE.
As such, it may be common to use MSE loss to train a regression predictive model, and to use RMSE to evaluate and report its performance.

The RMSE can be calculated as follows:

- RMSE = sqrt(1 / N * sum for i to N (y_i – yhat_i)^2)

Where *y_i* is the i'th expected value in the dataset, *yhat_i* is the i'th predicted value, and *sqrt()* is the square root function.

We can restate the RMSE in terms of the MSE as:

- RMSE = sqrt(MSE)

Note that the RMSE cannot be calculated as the average of the square root of the mean squared error values. This is a common error made by beginners and is an example of Jensen's inequality.
You may recall that the square root is the inverse of the square operation. MSE uses the square operation to remove the sign of each error value and to punish large errors. The square root reverses this operation, although it ensures that the result remains positive.

The root mean squared error between your expected and predicted values can be calculated using the mean_squared_error() function from the scikit-learn library.
By default, the function calculates the MSE, but we can configure it to calculate the square root of the MSE by setting the "*squared*" argument to *False*.

## Mean Absolute Error

Mean Absolute Error, or MAE, is a popular metric because, like RMSE, the units of the error score match the units of the target value that is being predicted. Unlike the RMSE, the changes in MAE are linear and therefore intuitive.

That is, MSE and RMSE punish larger errors more than smaller errors, inflating or magnifying the mean error score. This is due to the square of the error value. The MAE does not give more or less weight to different types of errors and instead the scores increase linearly with increases in error.

As its name suggests, the MAE score is calculated as the average of the absolute error values. Absolute or *abs()* is a mathematical function that simply makes a number positive. Therefore, the difference between an expected and

predicted value may be positive or negative and is forced to be positive when calculating the MAE.

The MAE can be calculated as follows:

- MAE = 1 / N * sum for i to N abs(y_i – yhat_i)

Where *y_i* is the i'th expected value in the dataset, *yhat_i* is the i'th predicted value and *abs()* is the absolute function.

# Model Hyperparameter Optimization

Machine learning models have hyperparameters.

Hyperparameters are points of choice or configuration that allow a machine learning model to be customized for a specific task or dataset.

- **Hyperparameter**: Model configuration argument specified by the developer to guide the learning process for a specific dataset.

Machine learning models also have parameters, which are the internal coefficients set by training or optimizing the model on a training dataset.

Parameters are different from hyperparameters. Parameters are learned automatically; hyperparameters are set manually to help guide the learning process.

A range of different optimization algorithms may be used, although two of the simplest and most common methods are random search and grid search.

- **Random Search**. Define a search space as a bounded domain of hyperparameter values and randomly sample points in that domain.
- **Grid Search**. Define a search space as a grid of hyperparameter values and evaluate every position in the grid.

Grid search is great for spot-checking combinations that are known to perform well generally.

Random search is great for discovery and getting hyperparameter combinations that you would not have guessed intuitively, although it often requires more time to execute.

Log-uniform is useful for searching penalty values as we often explore values at different orders of magnitude

Poisson regression, also known as a log-linear model, is what you use when your outcome variable is a count (i.e., numeric, but not quite so wide in range as a continuous variable.) Examples of count variables in research include how many heart attacks or strokes one's had, how many days in the past month one's used [insert your favorite illicit substance here], or, as in survival analysis, how many days from outbreak until infection.

## Generalized Additive Models

- LinearGAM.
- LogisticGAM.
- GammaGAM.
- PoissonGAM.

GAM is a model which allows the linear model to learn nonlinear relationships. It assumes that instead of using simple weighted sums it can use the sum of arbitrary functions.

GAM(Generalized Additive Model) is an extension of linear models.

If the data is having a nonlinear effect, in such a case we use GAM. Linearity in models means that the changes of one unit in predictors can cause the same effect on the outcome of the model. If at some point, changes in feature not affecting the outcome or impacting oppositely, we can say that there is a nonlinearity effect in the data.

In this situation, we can model relationships using one of the following techniques.

A simple transformation of the features.

Categorization of the feature.

Generalized additive models (GAMs).

GAM is a model which allows the linear model to learn nonlinear relationships. It assumes that instead of using simple weighted sums it can use the sum of arbitrary functions of each variable to model the outcome.

The formula of GAM can be represented as:

$$g(EY(y|x))=\beta 0+f1(x1)+f2(x2)+\ldots+fp(xp)$$

It is pretty similar to the formula of the regression model but instead of using BiXi (simple weighted sum), it uses f1(X1) (flexible function). In the core, it is still the sum of feature effects. Instead of modelling all relationships, we can also choose some features for modelling relationships because it supports the linear effect also.

Splines are functions that can be used in order to learn arbitrary functions. The spline function can make a variety of shapes to model the relationship in a better way.