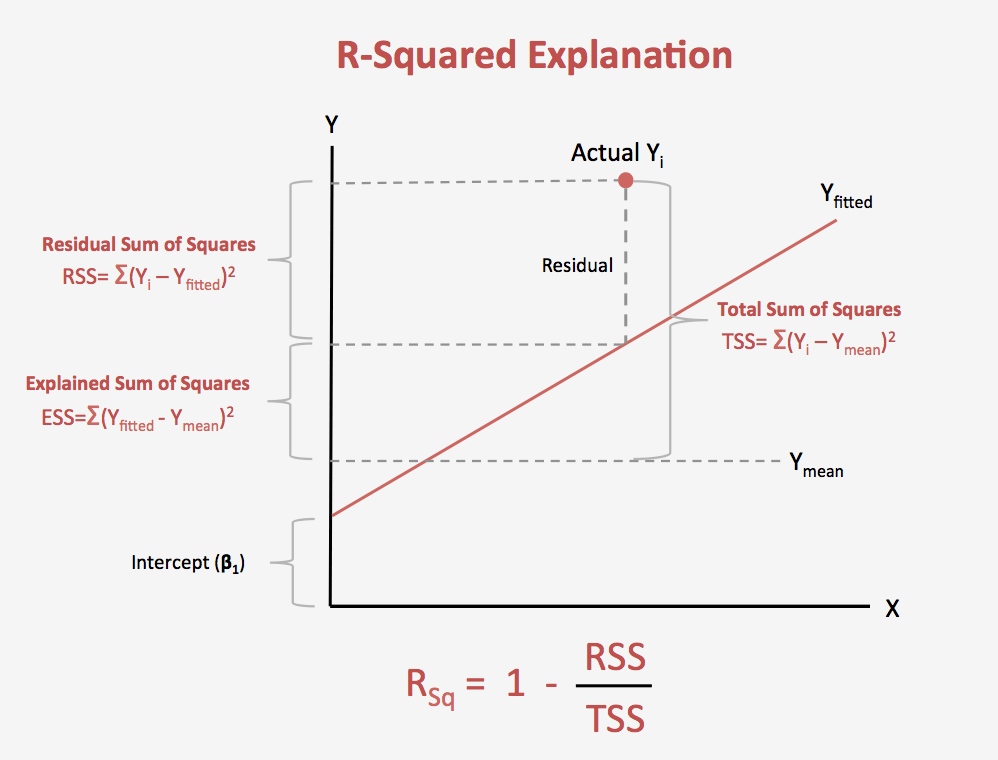
# INTRODUCTION

In statistical modeling, regression analysis is a set of statistical processes for estimating the relationships among variables. It includes many techniques for modeling and analyzing several variables, when the focus is on the relationship between a dependent variable (or DV / target/ response/ Y) and one or more independent variables (or IV / predictors / explanatory / X).

Linear regression is a linear approach to modelling the relationship between a dependent variable and one or more explanatory variables (or independent variables).



## SIMPLE LINEAR REGRESSION

Simple linear regression is a linear regression model with a single explanatory and single dependent variable and finds a linear function that as accurately as possible predicts the dependent variable values as a function of the independent variables.

## MULTIPLE LINEAR REGRESSION

Multiple linear regression is a form of regression analysis in which the relationship between multiple independent variable and single dependent variable is determined.

## POLYNOMIAL REGRESSION

Polynomial regression is a form of regression analysis in which the relationship between the independent variable x and the dependent variable y is modelled as an nth degree polynomial in x.

Least-squares problems fall into two categories depending on whether or not the residuals are linear in all unknowns:

* Linear least squares
* Nonlinear least squares

Polynomial least squares describe the variance in a prediction of the dependent variable as a function of the independent variable and the deviations from the fitted curve.

For linear models, the least squares minimization is usually done analytically using calculus. For nonlinear models, on the other hand, the minimization must almost always be done using iterative numerical algorithms.

## RANSAC LINEAR REGRESSION

RANdom SAmple Consensus (RANSAC) is an iterative method to make any parameter estimator strong against outliers. For example, of line fitting, RANSAC enable to estimate a line parameter even though data points include wrong point observations far from the true line.

RANSAC is composed of two steps:

* Hypothesis generation.
* Hypothesis evaluation.

In the first step, RANSAC estimates a line (hypothesis) from randomly sampled point data. In the second step, RANSAC counts the number of points which support the estimated line within the given threshold.

After several iterations of two steps, the final line can be obtained as a line who got the most supporters and inliers can be determined as its supporters.

## GENERALIZED LINEAR MODELS

The generalized linear model can be used to predict responses for dependent variables with discrete distributions and for dependent variables which are nonlinearly related to the predictors. For example, health vs age for people does not follow a linear relationship hence we can not predict heath status of a patient based on linear approach. We can use a GLM with a link function which will define the relation between age and health status using some function.

To illustrate, in the general linear model a response variable Y is linearly associated with values on the X variables by



where ‘e’ stands for the error variability that cannot be accounted for by the predictors; note that the expected value of e is assumed to be 0)

While the relationship in the generalized linear model is assumed to be



where e is the error, and g (…) is a function.

Formally, the inverse function of g (…), say f (…), is called the link function; so that:

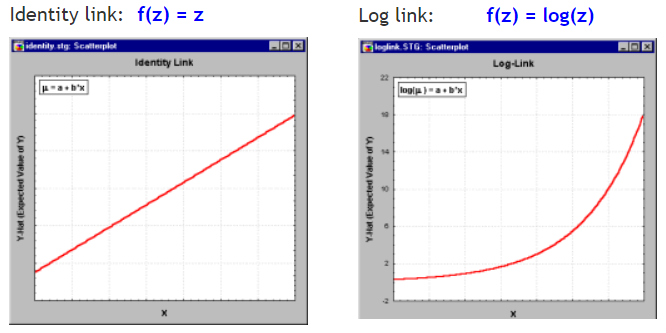


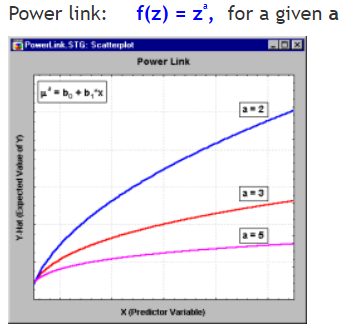
where muy stands for the expected value of y.

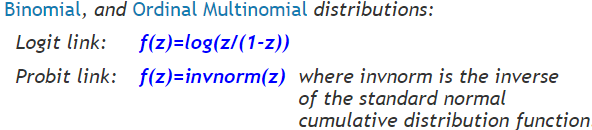
Further reading: <http://www.statsoft.com/Textbook/Generalized-Linear-Models>

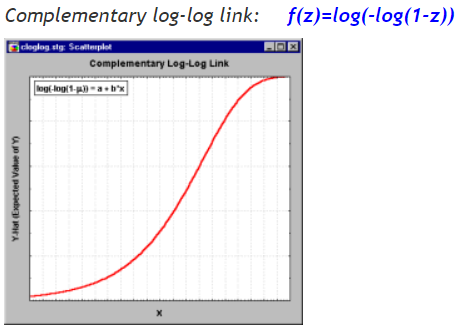
## LINK FUNCTIONS AND DISTRIBUTIONS

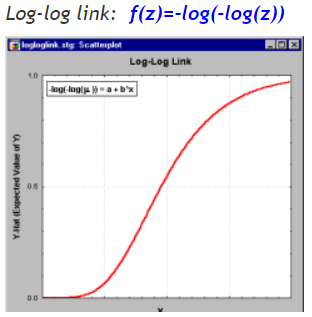
Various link functions can be chosen, depending on the assumed distribution of the y variable values:





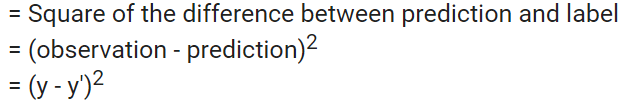


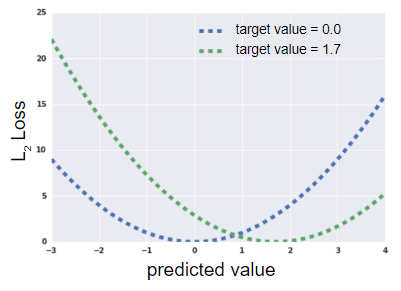




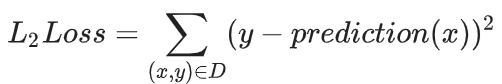
## LOSS FUNCTION

One of the commonly used loss functions for a regression problem is L2 Loss function also known as **squared loss**. It is defined as the square of difference between prediction and label.





Defining loss on the whole dataset



## COST FUNCTION

Some of the commonly used cost functions for regression analysis are:

* Mean Squared Error (MSE)
* Mean Absolute Error (MAE)
* Mean Squared Log Error

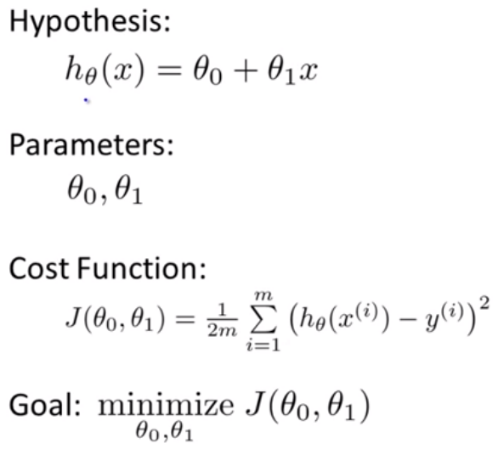
## OBJECTIVE FUNCTION

Commonly used objective functions to minimize the squared loss are:

* Ordinary Least Squares
* Weighted Least Squares

# LINEAR REGRESSION

In supervised learning, we have a dataset known as training dataset which is feed to the learning algorithm which outputs a function



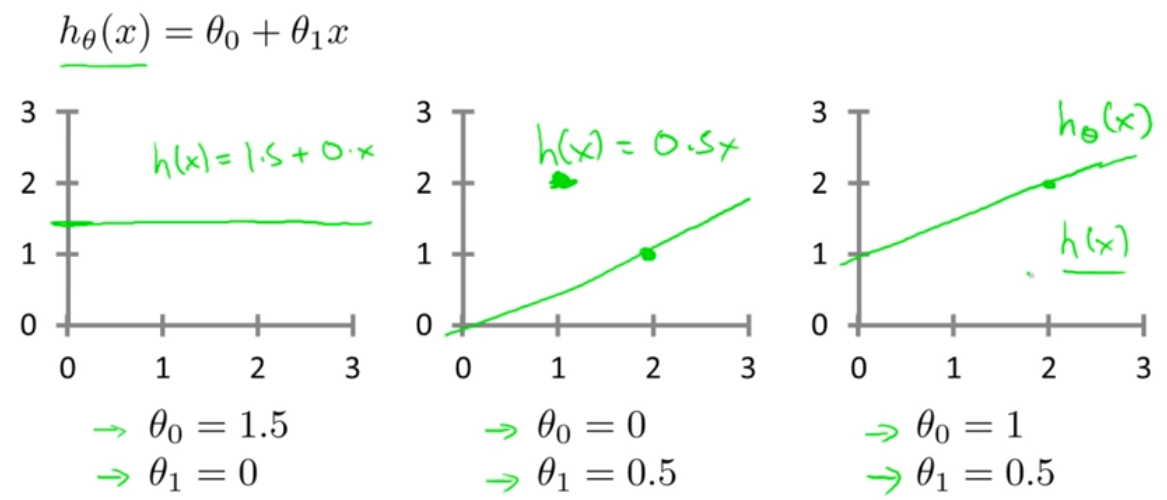
Where

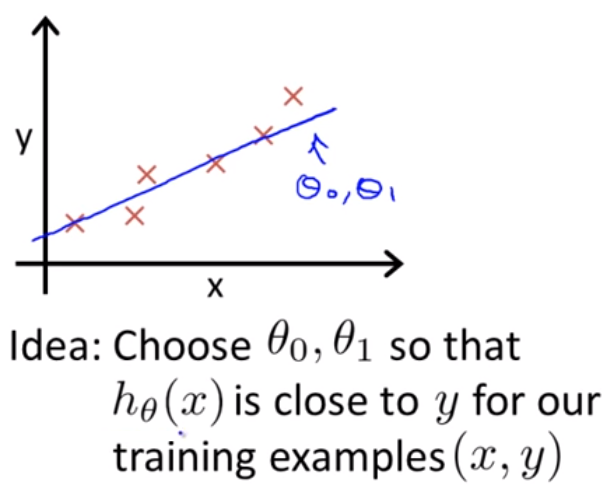
* m = no of training examples
* x's = input variable
* y = output variable
* h = output function (hypothesis)

So, h is a function which maps x's to y.

* (x, y) is one training example
* (xi, yi) is the ith row in that training set

h is a linear function with one variable x hence called univariate linear regression.





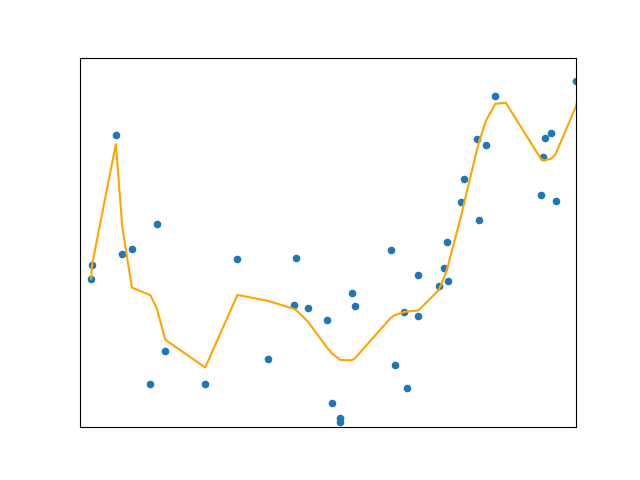
Objective function = Minimize the difference between predicted and observed dependent values

Find the values of Φ0 and Φ1 so that the average of sum squared errors is minimized. This function is known as cost function (also known as squared error function).

We Usually take the square of the error’s coz squared error cost functions works pretty well with regressions problems compared with other cost functions.

# NON-LINEAR REGRESSION

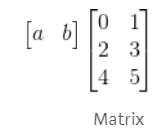
Linear regression requires the relation between the dependent variable and the independent variable to be linear. What if the distribution of the data was more complex as shown in the below figure? Can linear models be used to fit non-linear data? How can we generate a curve that best captures the data as shown below?



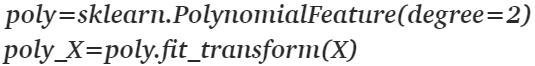
To convert the original features into their higher order terms we will use the PolynomialFeatures class provided by scikit-learn.

PolynomialFeatures take a matrix of features and transforms it into a feature matrix of quadratic nature (in case of degree two).

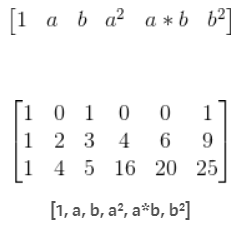
Let’s say we have a matrix of two features



Now we apply



What we actually get is a matrix of below form



# REGRESSION MODELS

We have different regression models for different datasets, some of the important ones are:

1. Ordinary Least Squares (OLS)
2. Weighted Least Squares (WLS)
3. Quantile Regression (QUANTREG)
4. Generalized Least Squares (GLS)
5. Recursive least squares (RecursiveLS)
6. Linear Mixed effect (mixedlm)
7. RANSAC Linear Regression.

If we use OLS then all the assumptions of linear regression must be met. One of the main assumptions being linear relationship between features and label and Homoscedasticity i.e. residuals have constant variance.

If residuals do not have constant variance i.e. data is heteroscedastic, then we use weighted least squares.

If we have Autocorrelation between residuals which usually happens with time series data, then we use generalized least square.

If we have dataset in which label varies differently with different combination of features values, then we should use quantile regression. For example, infant weights where mother is a smoker or non- smoker, mother bmi values etc.

Recursive least squares is used to find optimum least squares weight. Recursive least square is used in scenarios where we get data overtime and the algorithm keep updating as it gets new data. It usually used in time series analysis.

# REGRESSION ALGORITHMS

Below are some of the most used models for regression problem:

1. Linear regression (fits a straight line) [continuous model]
2. Polynomial regression (fits a curve) [continuous model]
3. Support Vector Regression (fits a line if kernel = linear else curve) [continuous model]
4. Decision tree regression (step graph) [non-continuous model]
5. Random forest regression (as per the data) [non-continuous model]

Regularization techniques:

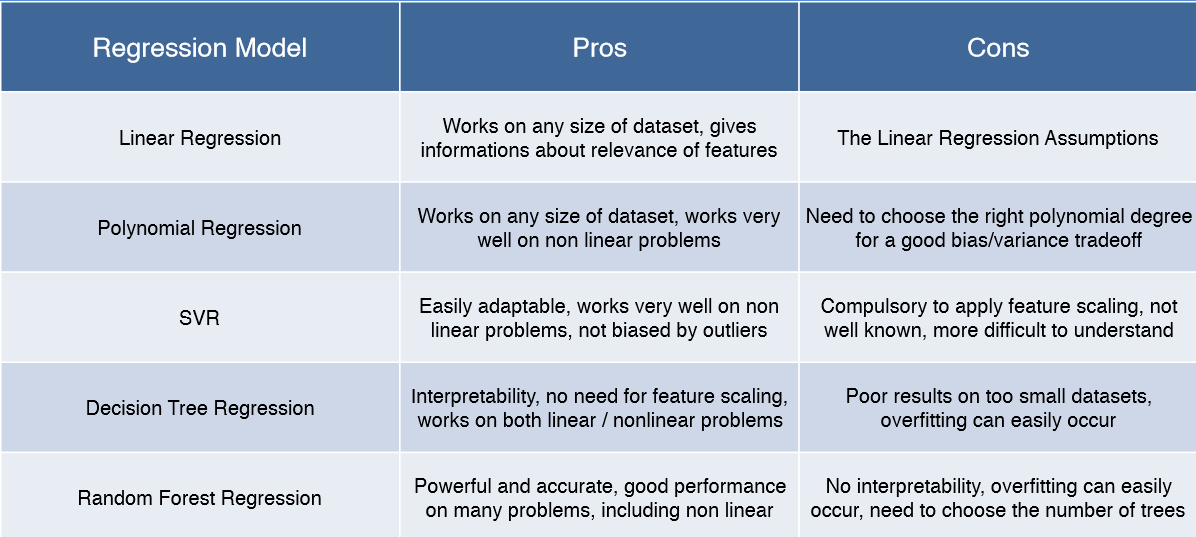
1. Lasso (L1 regularizer)
2. Ridge (L2 regularizer)
3. Elastic Net (L1 & L2 regularizer)
4. Stochastic Gradient Descent (SGD) Regressor

**If our problem is linear, you should go for Simple Linear Regression. If our problem is nonlinear, we should go for Polynomial Regression, SVR, Decision Tree or Random Forest.**

Then which one should you choose among these four? The method consists of using a very relevant technique that evaluates your model’s performance, called k-Fold Cross Validation, and then picking the model that shows the best results.

For SVR model we have to specifies the kernel type to be used in the algorithm. It must be one of ‘linear’, ‘poly’, ‘rbf’. If none is given, ‘rbf’ will be used.

Below are the pros and cons of each model



# COEFFICIENT CALCULATION

For Regression model we try to minimize RMSE by choosing appropriate coefficients weights. We have below methods to minimize RMSE:

1. Ordinary Least Squares
2. Weighted Least Squares
3. Generalized Least Squares
4. Quantile Least Squares

## ORDINARY LEAST SQUARES

In statistics, ordinary least squares (OLS) is a type of linear least squares method for estimating the unknown parameters in a linear regression model. OLS chooses the parameters of a linear function of a set of explanatory variables by the principle of least squares: minimizing the sum of the squares of the differences between the observed dependent variable (values of the variable being predicted) in the given dataset and those predicted by the linear function.

## WEIGHTED LEAST SQUARES

Weighted Least Squares is an extension of Ordinary Least Squares regression. Non-negative constants (weights) are attached to data points. It is used when any of the following are true:

One of OLS’ major assumptions is that the data should be linear and additive and therefore, the residuals are homoscedastic. OLS can handle a certain level of heteroskedasticity. One rules-of-thumb is like “the highest variability shouldn’t be greater than four times that of the smallest”.

The truth of the matter is, lots of data exhibits this “Heteroskedasticity”. Draw up some example feature-response relationships and we can often intuitively explain why:

* As age increases, net worth’s tend to diverge.
* As company size increases, revenues tend to diverge.
* Or, as infant height increases, weight tends to diverge.

In such cases we can use weighted least squares method to calculate the coefficients to minimize the errors. Use WLS when outliers are not all considered equal!

* Your data violates the assumption of homoscedasticity. In simple terms this means that your dependent variable should be clustered with similar variances, creating an even scatter pattern. If your data doesn’t have equal variances, you shouldn’t use OLS.
* You want to concentrate on certain areas (like low input value). OLS can’t “target” specific areas, while weighted least squares works well for this task. You may want to highlight specific areas in your study: ones that might be costly, expensive or painful to reproduce. By giving these areas bigger weights than others.
* You’re running the procedure as part of logistic regression or some other nonlinear function. With any non-linear procedure, linear regression is usually not the most appropriate modeling tool unless you can group the data. In addition, the error terms in logistic regression are heteroscedastic, which means you can’t use OLS.
* You have any other situation where data points should not be treated equally. For example, you might give more preference to points you know have been precisely measured and a lower preference to points that are estimated.

## COEFFICIENTS INTERPRETATION

Regression coefficients are estimates of the unknown population parameters and describe the relationship between a predictor variable and the response.

In linear regression, coefficients are the values that multiply the predictor values. Suppose you have the following regression equation: y = 3X + 5. In this equation, +3 is the coefficient, X is the predictor, and +5 is the constant.

The sign of each coefficient indicates the direction of the relationship between a predictor variable and the response variable.

* A positive sign indicates that as the predictor variable increases, the response variable also increases.
* A negative sign indicates that as the predictor variable increases, the response variable decreases.

The coefficient value represents the mean change in the response given a one-unit change in the predictor. For example, if a coefficient is +3, the mean response value increases by 3 for every one-unit change in the predictor.

P-values and coefficients in regression analysis work together to tell you which relationships in your model are statistically significant and the nature of those relationships. The coefficients describe the mathematical relationship between each independent variable and the dependent variable. The p-values for the coefficients indicate whether these relationships are statistically significant.

In linear regression the coefficients are determined in such a way that the error term between the IVs and DV is minimised.

# ASSUMPTIONS

For building a successful regression model it’s essential to validate below assumptions about the data that will be used to train the algorithm:

1. There should be a linear and additive relationship between dependent (response) variable and independent (predictor) variable(s).

A linear relationship suggests that a change in response Y due to one-unit change in X¹ is constant, regardless of the value of X¹.

An additive relationship suggests that the effect of X¹ on Y is independent of other variables.

1. There should be no correlation between the residual (error) terms. Absence of this phenomenon is known as Autocorrelation.
2. The independent variables should not be correlated. Absence of this phenomenon is known as multicollinearity.
3. The error terms (residuals) must have constant variance. This phenomenon is known as homoscedasticity. The presence of non-constant variance is referred to heteroscedasticity.
4. The error terms (residuals) must be normally distributed.

Let’s dive into specific assumptions and learn about their outcomes if violated.

## LINEAR AND ADDITIVE

Independent variables and dependent variable should have linear relationship.

If we fit a linear model to a non-linear, non-additive data set, the regression algorithm would fail to capture the trend mathematically, thus resulting in an inefficient model. Also, this will result in inaccurate predictions on an unseen data set.

CHECK: Check the scatterplots between the DV (Y) and each of the IVs (X) to determine linearity.

* Are there any bivariate outliers? If so, consider removing the outliers.
* Are there any non-linear relationships? If so, consider using a more appropriate type of regression.

## AUTOCORRELATION

Error terms should not be correlated with each other and should follow a normal distribution. Data points in the Residual plot should not have a visible pattern (autocorrelation).The data points in QQ plot should be symmetric around the line (normal distribution).

The presence of correlation in error terms drastically reduces model’s accuracy. This usually occurs in time series models where the next instant is dependent on previous instant. If the error terms are correlated, the estimated standard errors tend to underestimate the true standard error.

If this happens, it causes confidence intervals and prediction intervals to be narrower. Narrower confidence interval means that a 95% confidence interval would have lesser probability than 0.95 that it would contain the actual value of coefficients.

For example, the least square coefficient of X¹ is 15.02 and its standard error is 2.08 (without autocorrelation). But in presence of autocorrelation, the standard error reduces to 1.20. As a result, the prediction interval narrows down to (13.82, 16.22) from (12.94, 17.10).

Also, lower standard errors would cause the associated p-values to be lower than actual. This will make us incorrectly conclude a parameter to be statistically significant.

CHECK: Look for Durbin – Watson (DW) statistic. It must lie between 0 and 4. If DW = 2, implies no autocorrelation, 0 < DW < 2 implies positive autocorrelation while 2 < DW < 4 indicates negative autocorrelation. We can see residual plot and look for the seasonal or correlated pattern in residual values.

## MULTICOLLINEARITY

Independent variables should not have correlation within themselves.

This phenomenon exists when the independent variables are found to be moderately or highly correlated. In a model with correlated variables, it becomes a tough task to figure out the true relationship of a predictors with response variable. In other words, it becomes difficult to find out which variable is actually contributing to predict the response variable.

Another point, with presence of correlated predictors, the standard errors tend to increase. And, with large standard errors, the confidence interval becomes wider leading to less precise estimates of slope parameters.

Also, when predictors are correlated, the estimated regression coefficient of a correlated variable depends on which other predictors are available in the model. If this happens, you’ll end up with an incorrect conclusion that a variable strongly / weakly affects target variable. Since, even if you drop one correlated variable from the model, its estimated regression coefficients would change. That’s not good!

CHECK: You can use scatter plot to visualize correlation effect among variables. Also, you can also use VIF factor. VIF value <= 4 suggests no multicollinearity whereas a value of >= 10 implies serious multicollinearity.

## HOMOSCEDASTICITY

The error terms should have equal variance. Check for funnel shape in the residual plot.

The presence of non-constant variance in the error terms results in heteroskedasticity. Generally, non-constant variance arises in presence of outliers or extreme leverage values. Look like, these values get too much weight, thereby disproportionately influences the model’s performance. When this phenomenon occurs, the confidence interval for out of sample prediction tends to be unrealistically wide or narrow.

CHECK: You can look at residual vs fitted values plot. If heteroscedasticity exists, the plot would exhibit a funnel shape pattern.

## NORMAL DISTRIBUTION OF RESIDUALS (ERROR TERMS)

If the error terms are non- normally distributed, confidence intervals may become too wide or narrow. Once confidence interval becomes unstable, it leads to difficulty in estimating coefficients based on minimization of least squares. Presence of non – normal distribution suggests that there are a few unusual data points which must be studied closely to make a better model.

CHECK: You can look at QQ plot (plot should fall along a line)

* Residuals are more likely to be normally distributed if each of the variables normally distributed, so check normality first.
* Scatterplot should have no pattern
* Histogram should be normally distributed

If residuals are not normally distributed, there is probably something wrong with the distribution of one or more variables - re-check

## CHECK VARIABLES FOR NORMAL DISTRIBUTION

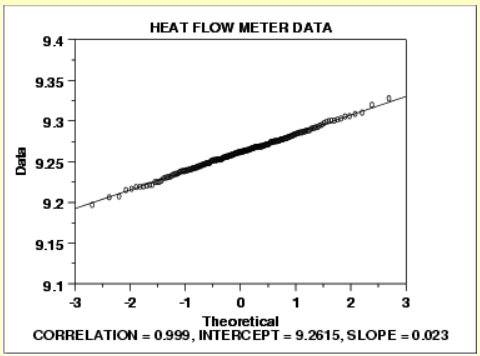
Normally distributed variables will enhance the MLR solution. Estimates of correlations will be more reliable and stable when the variables are normally distributed, but regression will be reasonably robust to minor to moderate deviations from non-normal data when moderate to large sample sizes are involved.

Also examine scatterplots for bivariate outliers because non-normal univariate data may make bivariate and multivariate outliers more likely.

CHECK: Check the histograms with a normal curve imposed.

# NORMAL PROBABILITY PLOT

To check the normal distribution of a feature, we can create a probability plot. Normally distributed data will have following plot:



We can make the following conclusions from the above plot.

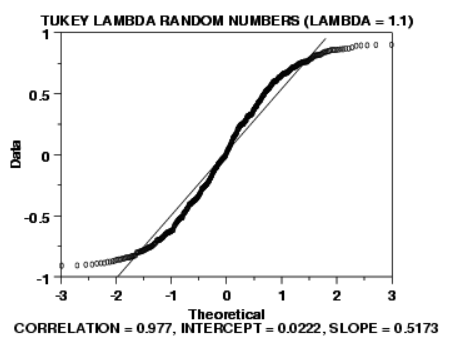
* The normal probability plot shows a strongly linear pattern. There are only minor deviations from the line fit to the points on the probability plot.
* The normal distribution appears to be a good model for these data.

In this case, we can quite reasonably conclude that the normal distribution provides an excellent model for the data.

## DATA HAVE SHORT TAILS

We can make the following conclusions from the above plot.

* The normal probability plot shows a non-linear pattern.
* The normal distribution is not a good model for these data.

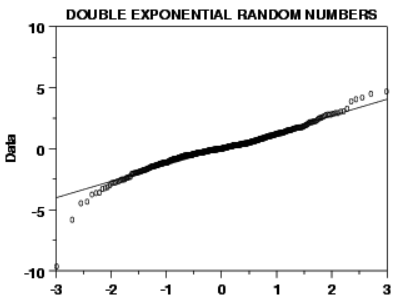


In this case, we can reasonably conclude that the normal distribution does not provide an adequate fit for this data set.

## DATA HAVE LONG TAILS

We can make the following conclusions from the above plot.

* The normal probability plot shows a reasonably linear pattern in the center of the data. However, the tails, particularly the lower tail, show departures from the fitted line.
* A distribution other than the normal distribution would be a good model for these data.

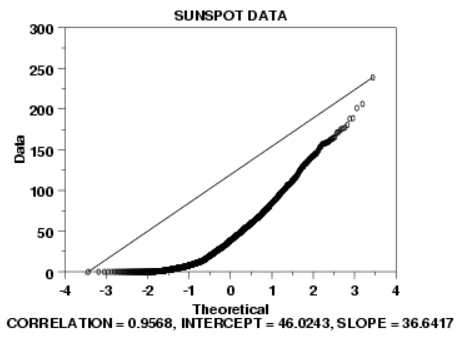


In this case we can reasonably conclude that the normal distribution can be improved upon as a model for these data.

## DATA ARE SKEWED RIGHT

We can make the following conclusions from the above plot.

* The normal probability plot shows a strongly non-linear pattern. Specifically, it shows a quadratic pattern in which all the points are below a reference line drawn between the first and last points.
* The normal distribution is not a good model for these data.



This quadratic pattern in the normal probability plot is the signature of a significantly right-skewed data set. Similarly, if all the points on the normal probability plot fell above the reference line connecting the first and last points, that would be the signature pattern for a significantly left-skewed data set.

In this case we can quite reasonably conclude that we need to model these data with a right skewed distribution such as the Weibull or lognormal.

# TESTING FOR MULTICOLLINEARITY WITH VIF

If you can identify which variables are affected by multicollinearity and the strength of the correlation, you’re well on your way to determining whether you need to fix it. Fortunately, there is a very simple test to assess multicollinearity in your regression model. The variance inflation factor (VIF) identifies correlation between independent variables and the strength of that correlation.

Statistical software calculates a VIF for each independent variable. VIFs start at 1 and have no upper limit. A value of 1 indicates that there is no correlation between this independent variable and any others. VIFs between 1 and 5 suggest that there is a moderate correlation, but it is not severe enough to warrant corrective measures. VIFs greater than 5 represent critical levels of multicollinearity where the coefficients are poorly estimated, and the p-values are questionable.

HOW TO DEAL WITH MULTICOLLINEARITY

Lower precision for regression coefficients (higher std error for coefficients), switched signs, and a lack of statistical significance (p-value) are typical problems associated with multicollinearity. Multicollinearity can be resolved using below steps:

1. **Centering the variables is a simple way to reduce structural multicollinearity.**

Centering the variables is also known as standardizing the variables by subtracting the mean. This process involves calculating the mean for each continuous independent variable and then subtracting the mean from all observed values of that variable. Then, use these centered variables in your model. Most statistical software provides the feature of fitting your model using standardized variables.

There are other standardization methods, but the advantage of just subtracting the mean is that the interpretation of the coefficients remains the same. The coefficients continue to represent the mean change in the dependent variable given a 1-unit change in the independent variable.

1. **Data (Non-Structural) Multicollinearity.**

There are a variety of methods that you can try, but each one has some drawbacks. You’ll need to use your subject-area knowledge and factor in the goals of your study to pick the solution that provides the best mix of advantages and disadvantages.

The potential solutions include the following:

* Remove some of the highly correlated independent variables.
* Linearly combine the independent variables, such as adding them together.
* Perform an analysis designed for highly correlated variables, such as principal components analysis or partial least squares regression.

# SOLUTION FOR AUTOCORRELATION

Autocorrelation in a model occurs when the residual terms are correlated with each other. It usually happens in time series data.

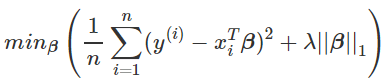
* Check using residual plot. If data is equally around 0 (means no pattern) then fine.
* This can be checked by looking at the Durbin Watson value. If = 2 then there is no auto correlation. Also, if lies between 1.5 & 2.5 we don’t have a serious problem and its acceptable. For our model its acceptable

If this occurs then we cannot use linear model for prediction. Instead use:

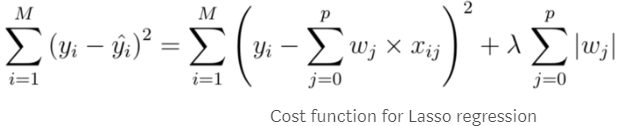
* Autoregressive Model.
* Moving Average Model.
* Autoregressive moving average Model.
* Autoregressive Integrated moving average Model.

# LASSO REGRESSION (L1)

Lasso Regression is quite similar to Ridge Regression in that both techniques have the same premise. We are again adding a biasing term to the regression optimization function in order to reduce the effect of collinearity and thus the model variance. However, instead of using a squared bias like ridge regression, lasso instead using an absolute value bias:



The cost function for Lasso (least absolute shrinkage and selection operator) regression can be written as



There are a few differences between the Ridge and Lasso regressions that essentially draw back to the differences in properties of the L2 and L1 regularization:

1. Built-in feature selection: is frequently mentioned as a useful property of the L1-norm, which the L2-norm does not. This is actually a result of the L1-norm, which tends to produces sparse coefficients. For example, suppose the model have 100 coefficients but only 10 of them have non-zero coefficients, this is effectively saying that “the other 90 predictors are useless in predicting the target values”. L2-norm produces non-sparse coefficients, so does not have this property. Thus, one can say that Lasso regression does a form of “parameter selections” since the feature variables that aren’t selected will have a total weight of 0.

L1 regularization may cause the following kinds of features to be given weights of exactly 0:

* Weakly informative features.
* Strongly informative features on different scales.
* Informative features strongly correlated with other similarly informative features.

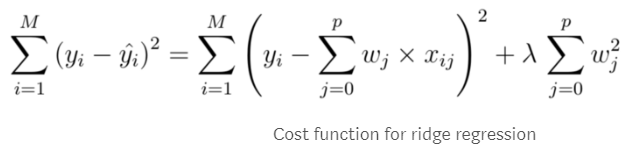
# RIDGE REGRESSION (L2)

A standard linear or polynomial regression will fail in the case where there is high collinearity among the feature variables. Collinearity is the existence of near-linear relationships among the independent variables. The presence of high collinearity can be determined in a few different ways:

* A regression coefficient is not significant even though, theoretically, that variable should be highly correlated with Y.
* When you add or delete an X feature variable, the regression coefficients change dramatically.
* Your X feature variables have high pairwise correlations (check the correlation matrix).

Above image shows ridge regression, where the 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.

In ridge regression, the cost function is altered by adding a penalty equivalent to square of the magnitude of the coefficients.



A few key points about Ridge Regression:

* The assumptions of this regression are same as least squared regression except normality is not to be assumed.
* It shrinks the value of coefficients but doesn’t reaches zero, which suggests no feature selection feature. Hence, we can perform feature selection and then pass the selected IVs.

# ELASTICNET REGRESSION (L1 & L2)

ElasticNet is a hybrid of Lasso and Ridge Regression techniques. It is uses both the L1 and L2 regularization taking on the effects of both techniques:

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.

A few key points about ElasticNet Regression:

* It encourages group effect in the case of highly correlated variables, rather than zeroing some of them out like Lasso.
* There are no limitations on the number of selected variables.

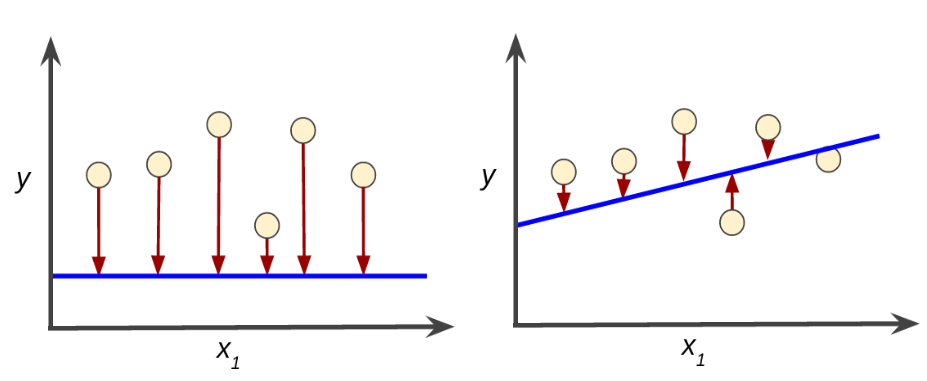
All of these regression regularization methods (Lasso, Ridge and ElasticNet) work well in case of high dimensionality and multicollinearity among the variables in the data set.

# TRAINING A MODEL

Training a model simply means learning (determining) good values for all the weights and the bias from labeled examples. In supervised learning, a machine learning algorithm builds a model by examining many examples and attempting to find a model that minimizes loss; this process is called empirical risk minimization.

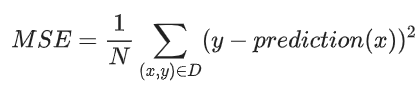
Loss is the penalty for a bad prediction. That is, loss is a number indicating how bad the model's prediction was on a single example. If the model's prediction is perfect, the loss is zero; otherwise, the loss is greater. The goal of training a model is to find a set of weights and biases that have low loss, on average, across all examples. For example, Figure 3 shows a high loss model on the left and a low loss model on the right. Note the following about the figure:

The arrows represent loss, the lines represent predictions.



## MEAN SQUARE ERROR

Mean square error (MSE) is the average squared loss per example over the whole dataset. To calculate MSE, sum up all the squared losses for individual examples and then divide by the number of examples:



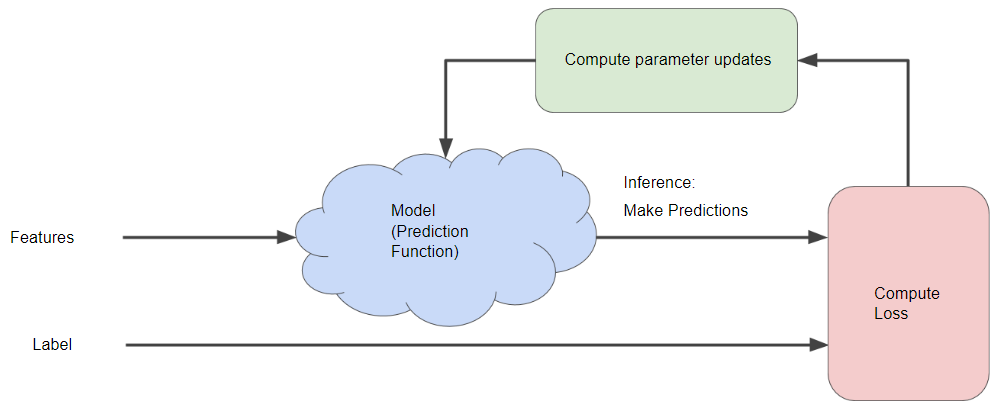
Although MSE is commonly-used in machine learning, it is neither the only practical loss function nor the best loss function for all circumstances.

## REDUCING LOSS

To train a model, we need a good way to reduce the model’s loss. An iterative approach is one widely used method for reducing loss, and is as easy and efficient as walking down a hill.

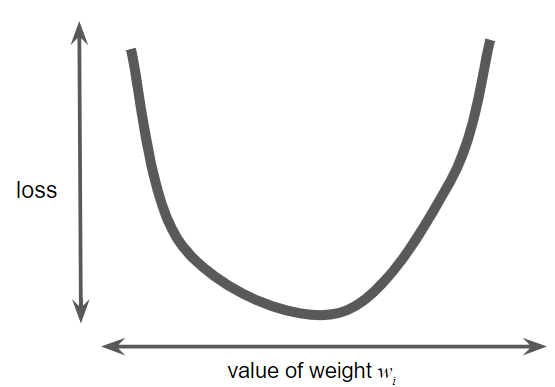
The "hidden object" is to find the best possible model. You'll start with a wild guess ("The value of w1 is 0.") and wait for the system to tell you what the loss is. Then, you'll try another guess ("The value of w1 is 0.5.") and see what the loss is.

The following figure suggests the iterative trial-and-error process that machine learning algorithms use to train a model:



The learning continues iterating until the algorithm discovers the model parameters with the lowest possible loss. Usually, you iterate until overall loss stops changing or at least changes extremely slowly. When that happens, we say that the model has converged.

Suppose we had the time and the computing resources to calculate the loss for all possible values of w1. For the kind of regression problems, we've been examining, the resulting plot of loss vs. w1 will always be convex. In other words, the plot will always be bowl-shaped, kind of like this:

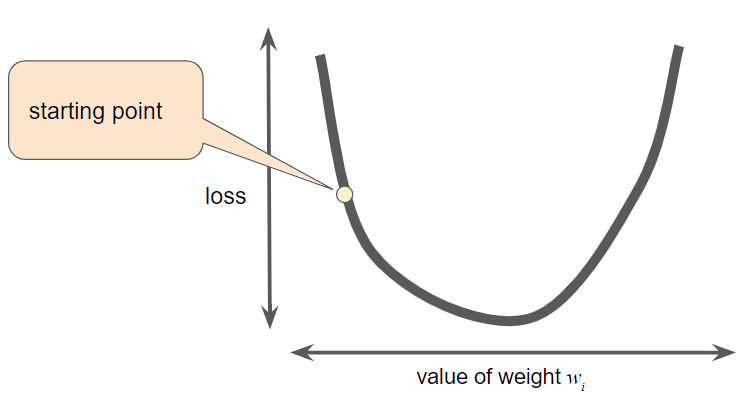


Convex problems have only one minimum; that is, only one place where the slope is exactly 0. That minimum is where the loss function converges.

Calculating the loss function for every conceivable value of w1 over the entire data set would be an inefficient way of finding the convergence point. Let's examine a better mechanism—very popular in machine learning—called gradient descent.

## GRADIENT DESCENT

The first stage in gradient descent is to pick a starting value (a starting point) for w1. The starting point doesn't matter much; therefore, many algorithms simply set w1 to 0 or pick a random value. The following figure shows that we've picked a starting point slightly greater than 0:

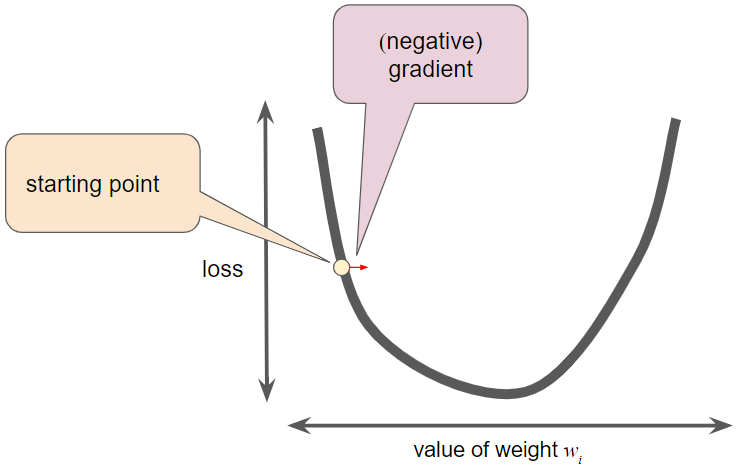


The gradient descent algorithm then calculates the gradient of the loss curve at the starting point. Here in Figure 3, the gradient of loss is equal to the derivative (slope) of the curve, and tells you which way is "warmer" or "colder." When there are multiple weights, the gradient is a vector of partial derivatives with respect to the weights.

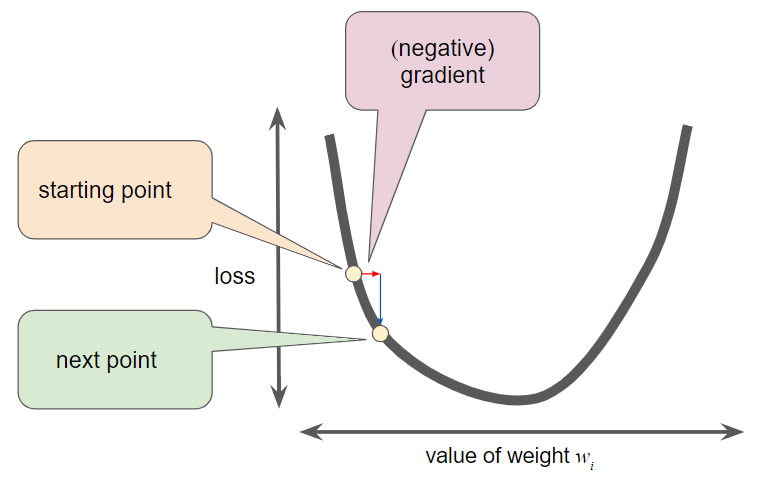
Note that a gradient is a vector, so it has both of the following characteristics:

* A direction
* A magnitude

The gradient always points in the direction of steepest increase in the loss function. The gradient descent algorithm takes a step in the direction of the negative gradient in order to reduce loss as quickly as possible.



To determine the next point along the loss function curve, the gradient descent algorithm adds some fraction of the gradient's magnitude to the starting point as shown in the following figure:



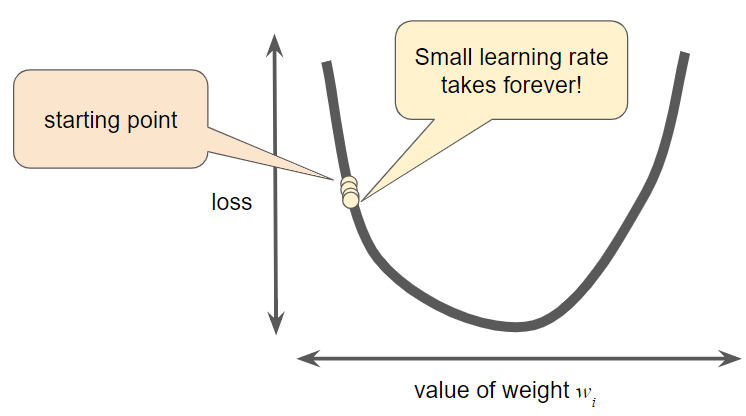
The gradient descent then repeats this process, edging ever closer to the minimum.

When performing gradient descent, we generalize the above process to tune all the model parameters simultaneously. For example, to find the optimal values of both w1 and the bias, we calculate the gradients with respect to both w1 and b. Next, we modify the values of w1 and b based on their respective gradients. Then we repeat these steps until we reach minimum loss.

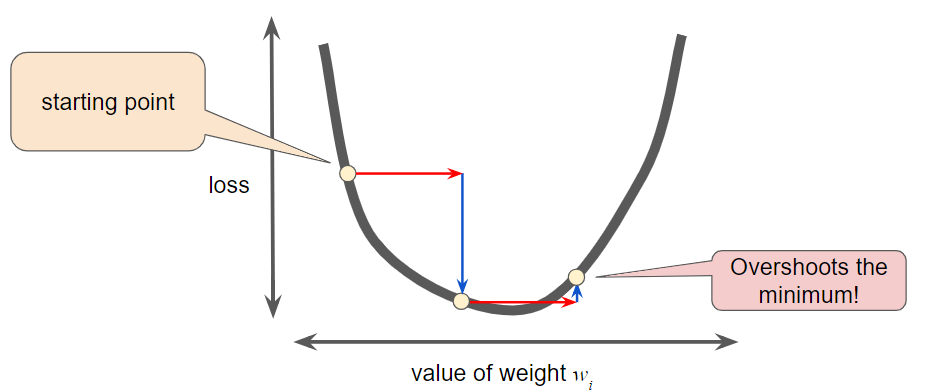
## LEARNING RATE

As noted, the gradient vector has both a direction and a magnitude. Gradient descent algorithms multiply the gradient by a scalar known as the learning rate (also sometimes called step size) to determine the next point. For example, if the gradient magnitude is 2.5 and the learning rate is 0.01, then the gradient descent algorithm will pick the next point 0.025 away from the previous point.

Hyperparameters are the knobs that programmers tweak in machine learning algorithms. Most machine learning programmers spend a fair amount of time tuning the learning rate. If you pick a learning rate that is too small, learning will take too long:



Conversely, if you specify a learning rate that is too large, the next point will perpetually bounce haphazardly across the bottom of the well like a quantum mechanics experiment gone horribly wrong:



In practice, finding a "perfect" (or near-perfect) learning rate is not essential for successful model training. The goal is to find a learning rate large enough that gradient descent converges efficiently, but not so large that it never converges.

# BIAS VARIANCE TRADE-OFF

Bias is calculated by finding the mean/ average of the error terms in the model whereas Variance is calculated by finding the standard deviations of the error terms for the model in Python.

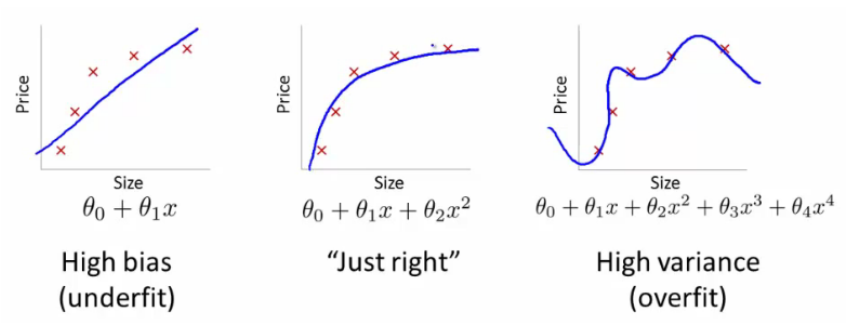
**Ideal Model:** Low Bias and Low Var

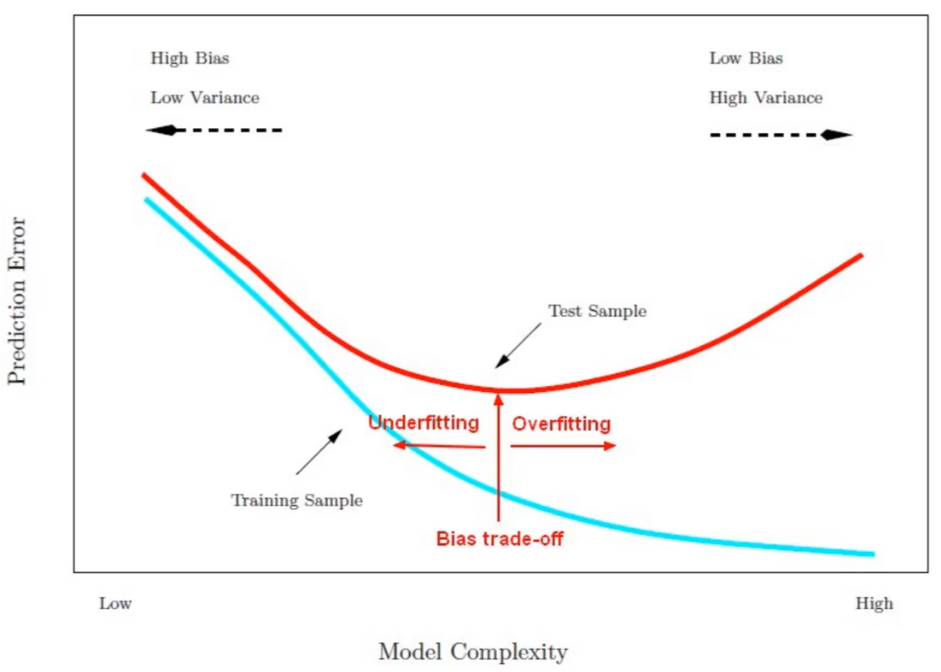
**Over fitting:** Low Bias & High Variance

Noise in the data hence model has learnt trends/ patters which are not true for the population.

**Under fitting:** High Bias & Low Variance

Dataset with less data so algorithm was not able to capture all the patterns.





## CHCEK FOR BIAS AND VARIANCE

Within model we can check for

* We can check validation and test error values
* We can check validation error vs error values at various levels while choosing different parameters (like no of clusters or no of neighbours).

Comparing models

* Bias and variance values
* AIC & BIC

## WHAT TO DO FOR OVERFITTING

Following are the commonly used methodologies:

1. Cross-Validation: Cross Validation in its simplest form is a one round validation, where we leave one sample as in-time validation and rest for training the model. But for keeping lower variance a higher fold cross validation is preferred.
2. Early Stopping: Early stopping rules provide guidance as to how many iterations can be run before the learner begins to over-fit.
3. Pruning: Pruning (trimming) is used extensively while building CART models. It simply removes the nodes which add little predictive power for the problem in hand.
4. Regularization: This is the technique we are going to discuss in more details. Simply put, it introduces a cost term for bringing in more features with the objective function. Hence, it tries to push the coefficients for many variables to zero and hence reduce cost term.

## WHAT TO DO FOR UNDERFITTING

1. Provide learning algorithm with additional data (more no of variables).
2. Use polynomial curve instead of linear equation to fit the data.

The rule of the thumb is:

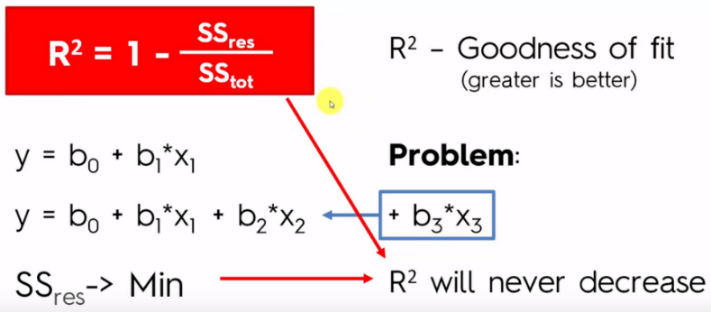
* If training error is high and cross validation error is also high it’s underfit.
* If the training error is low and cross validation error is high then it’s over fit.

# R-SQUARED (COEFFICIENT OF DETERMINATION)

R-squared is a statistical measure of how close the data are to the fitted regression line. It is also known as the coefficient of determination, or the coefficient of multiple determination for multiple regression. R-squared is less than or equal to 1 (can be negative).

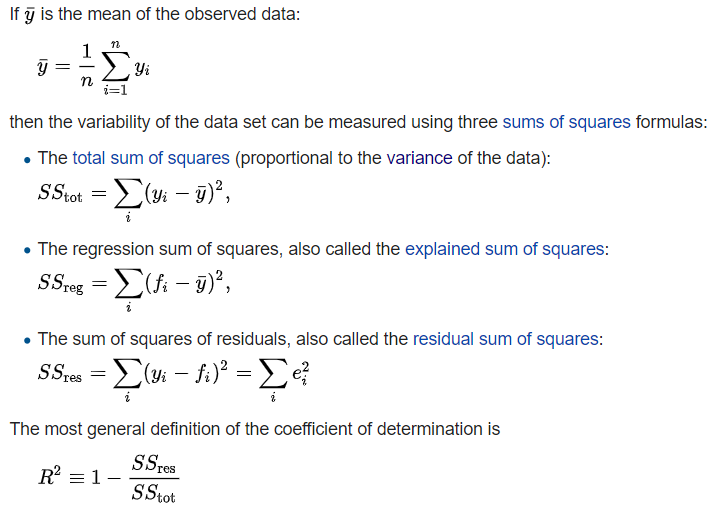
The definition of R-squared is fairly straight-forward: It is the percentage of the response variable (DV) variation that is explained by a linear model.

R-squared = Explained variation / Total variation

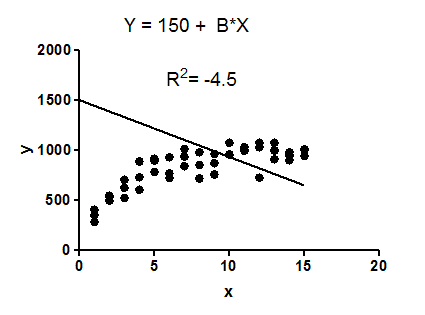


In general, the higher the R-squared, the better the model fits your data. However, there are important conditions for this guideline.

Formula for Total Sum of Squares, explained sum of squares, Residual Sum of squares are as below:



R2 compares the fit of the chosen model with that of a horizontal straight line (the null hypothesis). If the chosen model fits worse than a horizontal line, then R2 is negative. Note that R2 is not always the square of anything, so it can have a negative value without violating any rules of math. R2 is negative only when the chosen model does not follow the trend of the data, so fits worse than a horizontal line.



## KEY LIMITATIONS OF R-SQUARED

R-squared cannot determine whether the coefficient estimates and predictions are biased, which is why you must assess the residual plots.

Every time you add a predictor to a model, the R-squared increases, even if due to chance alone. It never decreases. Consequently, a model with more terms may appear to have a better fit simply because it has more terms.

If a model has too many predictors and higher order polynomials, it begins to model the random noise in the data. This condition is known as overfitting the model and it produces misleadingly high R-squared values and a lessened ability to make predictions.

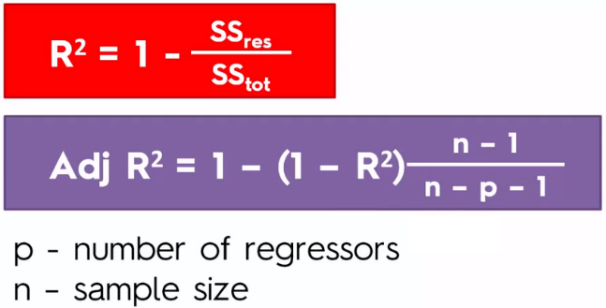




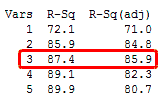
# ADJUSTED R-SQUARED

Suppose you compare a five-predictor model with a higher R-squared to a one-predictor model. Does the five-predictor model have a higher R-squared because it’s better? Or is the R-squared higher because it has more predictors? Simply compare the adjusted R-squared values to find out.

The adjusted R-squared is a modified version of R-squared with penalizing factor that has been adjusted for the number of predictors in the model. The adjusted R-squared increases only if the new term improves the model more than would be expected by chance. It decreases when a predictor improves the model by less than expected by chance. The adjusted R-squared can be negative, but it’s usually not. It is always lower than the R-squared.



In the simplified Best Subsets Regression output below, you can see where the adjusted R-squared peaks, and then declines. Meanwhile, the R-squared continues to increase.



# PREDICTED R-SQUARED

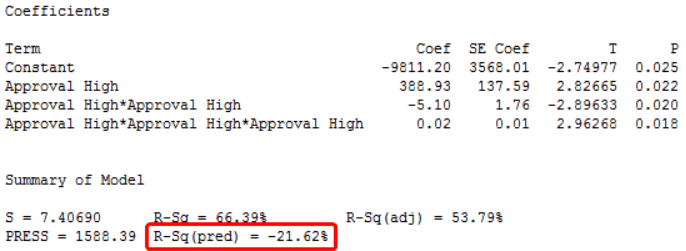
The predicted R-squared indicates how well a regression model predicts responses for new observations. This statistic helps you determine when the model fits the original data but is less capable of providing valid predictions for new observations.

A key benefit of predicted R-squared is that it can prevent you from overfitting a model. As mentioned earlier, an overfit model contains too many predictors and it starts to model the random noise.

Because it is impossible to predict random noise, the predicted R-squared must drop for an overfit model. If you see a predicted R-squared that is much lower than the regular R-squared, you almost certainly have too many terms in the model.

The predicted R-squared doesn’t have to be negative to indicate an overfit model. If you see the predicted R-squared start to fall as you add predictors, even if they’re significant, you should begin to worry about overfitting the model.

Let’s take an example:



In this example, both the R-squared and adjusted R-squared look pretty good! Also, the coefficient estimates are all significant because their p-values are less than 0.05. The residual plots (not shown) look good too. Great!

Our model is too complicated and the predicted R-squared gives this away. We actually have a negative predicted R-squared value. That may not seem intuitive, but if 0% is terrible, a negative percentage is even worse!

# CLOSING THOUGHTS (ADJUSTED R-SQUARED & PREDICTED R-SQUARED)

All data contain a natural amount of variability that is unexplainable. Unfortunately, R-squared doesn’t respect this natural ceiling. Chasing a high R-squared value can push us to include too many predictors in an attempt to explain the unexplainable.

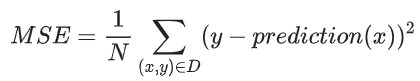
In these cases, you can achieve a higher R-squared value, but at the cost of misleading results, reduced precision, and a lessened ability to make predictions.

Both adjusted R-squared and predicted R-square provide information that helps you assess the number of predictors in your model:

* Use the adjusted R-square to compare models with different numbers of predictors.
* Use the predicted R-square to determine how well the model predicts new observations and whether the model is too complicated.

# ROOT MEAN SQUARED ERROR

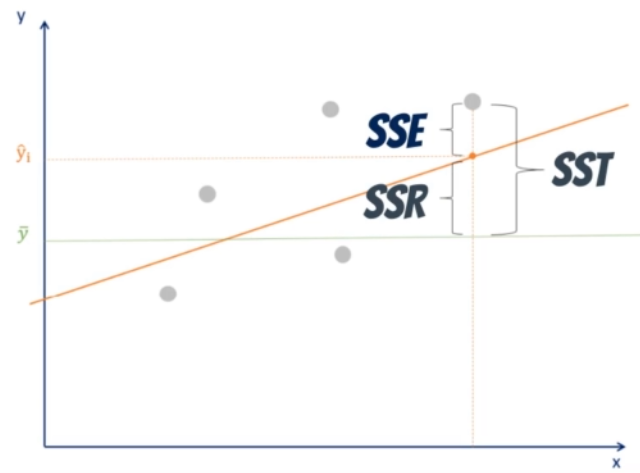
Mean square error (MSE) is the average squared loss per example over the whole dataset. To calculate MSE, sum up all the squared losses for individual examples and then divide by the number of examples:



RMSE is a measure of accuracy, to compare forecasting errors of different models for a particular dataset and not between datasets, as it is scale-dependent.

RMSE is always non-negative, and a value of 0 (almost never achieved in practice) would indicate a perfect fit to the data. In general, a lower RMSE is better than a higher one.

RMSE is a measure of how spread out these residuals are. In other words, it tells you how concentrated the data is around the line of best fit. Root mean square error is commonly used in forecasting and regression analysis to verify experimental results.



# TRAINING VS VALIDATION ERROR

The training loss is calculated over the entire training dataset. Likewise, the validation loss is calculated over the entire validation dataset.

The training set is typically at least 4 times as large as the validation (80-20). Given that the error is calculated over all samples, you could expect up to approximately 4X the loss measure of the validation set. You will notice, however, that the training loss and validation loss are approaching one another as training continues.

* **UNDER FITTING:** Train error & Validation error will be high
* **OVERFITTING:** Train error will be low & Validation error will be high
* **GOOD FIT:** Train and validation error approx. equal (validation little more than train)
* **UNKNOWN FIT:** Train error more than Validation error.

We 'unknown' fit because the result is counter intuitive to how machine learning works. The essence of ML is to predict the unknown. If you are better at predicting the unknown than what you have 'learned', AFAIK the data between training and validation must be different in some way. This could mean you either need to re-evaluate your data splitting method, adding more data, or possibly changing your performance metric.

Sometimes when we use regularization techniques it is possible that we might get train error more than train error. That will be because of no of iterations taken before model converges in training model.

# AKAIKE INFORMATION CRITERION

Akaike information criterion (AIC) is a model selection technique based on in-sample fit to estimate the likelihood of a model to predict/estimate the future values. A good model is the one that has minimum AIC among all the other models.

The Akaike information criterion (AIC), is generally considered the first model selection criterion that should be used in practice. The AIC is The AIC in isolation is meaningless. Rather, this value is calculated for every candidate model and the “best” model is the candidate model with the smallest AIC.

AIC is given by using below formula:



where L is the value of the likelihood, N is the number of recorded measurements, and k is the number of estimated parameters.

# BAYESIAN INFORMATION CRITERION

Bayesian information criterion (BIC) is another technique for model selection that measures the trade-off between model fit and complexity of the model. A lower AIC or BIC value indicates a better fit.



When comparing the Bayesian Information Criteria and the Akaike’s Information Criteria, penalty for additional parameters is more in BIC than AIC.

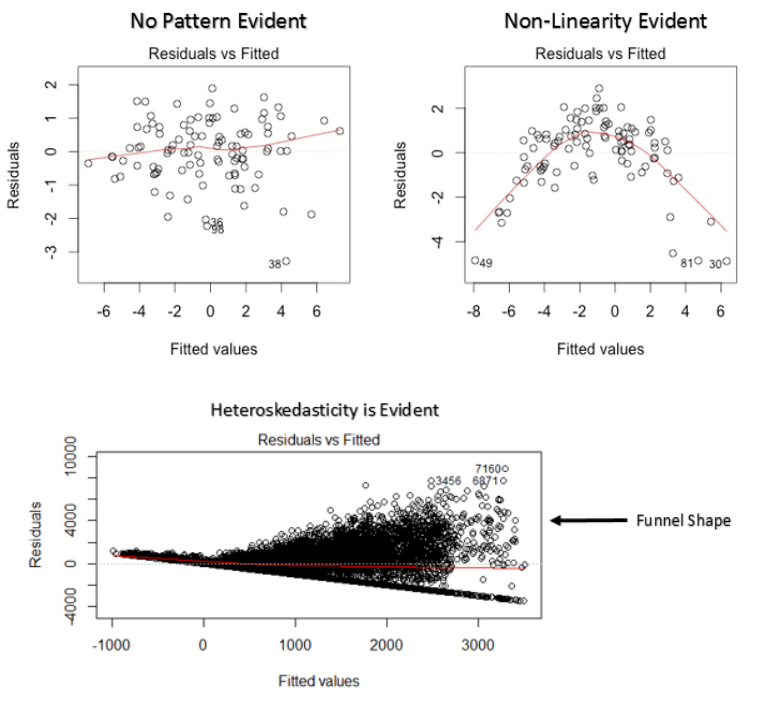
In the case of multivariate regression analysis, AIC is better than BIC in model selection. The optimal model is selected based on the highest R2 and minimum AIC and BIC.

# RESIDUAL PLOT

If the model fit to the data were correct, the residuals would approximate the random errors that make the relationship between the explanatory variables and the response variable a statistical relationship. Therefore, if the residuals appear to behave randomly, it suggests that the model fits the data well. On the other hand, if non-random structure is evident in the residuals, it is a clear sign that the model fits the data poorly.

Residual plots display the residual values on the y-axis and fitted values or another variable, on the x-axis.

Scatter plots of the residuals versus the predictor variables in the model and versus potential predictors that are not included in the model are the primary plots used to assess sufficiency of the functional part of the model. Plots in which the residuals do not exhibit any systematic structure indicate that the model fits the data well. Plots of the residuals versus other predictor variables, or potential predictors, that exhibit systematic structure indicate that the form of the function can be improved in some way.



If a funnel shape is evident in the plot, consider it as the signs of non-constant variance i.e. heteroskedasticity. This indicates that the model is not able to fit the data well and hence model needs to be modified.

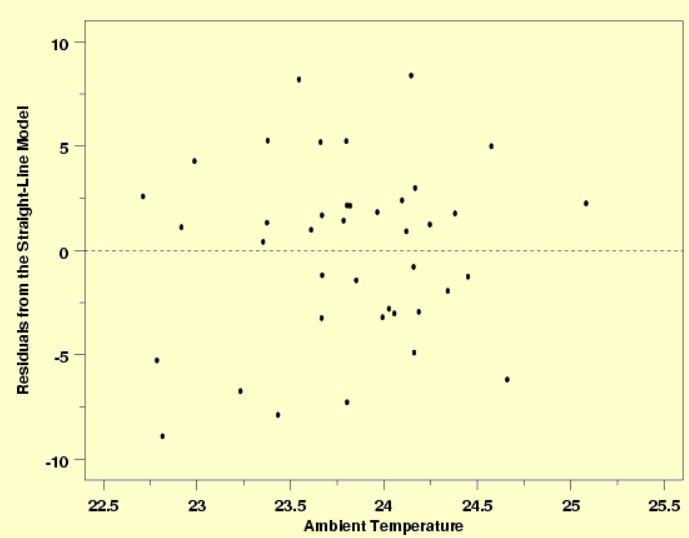
To overcome heteroskedasticity, a possible way is to transform the response variable such as log(Y) or √Y. Also, you can use weighted least square method to tackle heteroskedasticity.

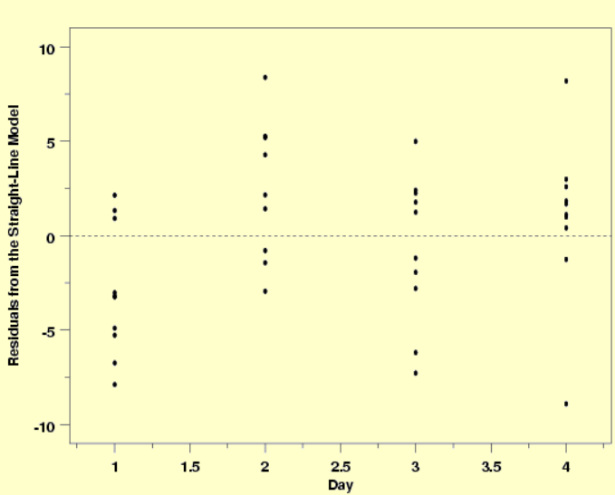
The above residual plot indicates that the independent variables do not capture the entire deterministic component. Unfortunately, some of the explanatory information has leaked over to the supposedly random error. There are a variety of reasons why a model can have this problem. The possibilities include a missing:

* Independent variable.
* Polynomial term to model a curve.
* Interaction term.

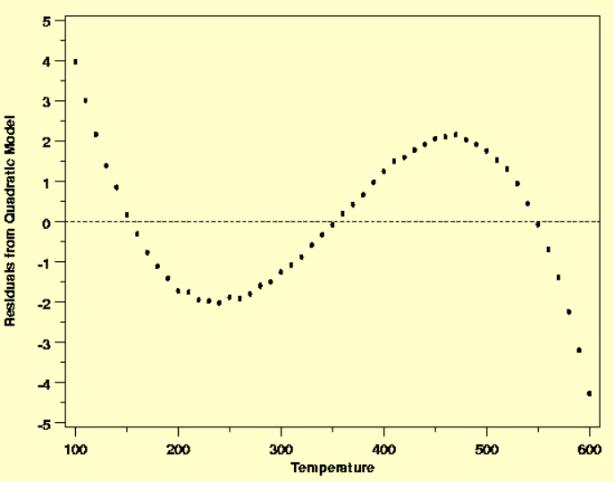
To fix the problem, you need to identify the missing information, variable, or higher-order term and include it in the model. After you correct the problem and refit the model, the residuals should look nice and random! It might require subject-area knowledge and research to do this. The solution is very particular to your research.

Below plots show that model fits the data well as no particular pattern is observed in the plot:





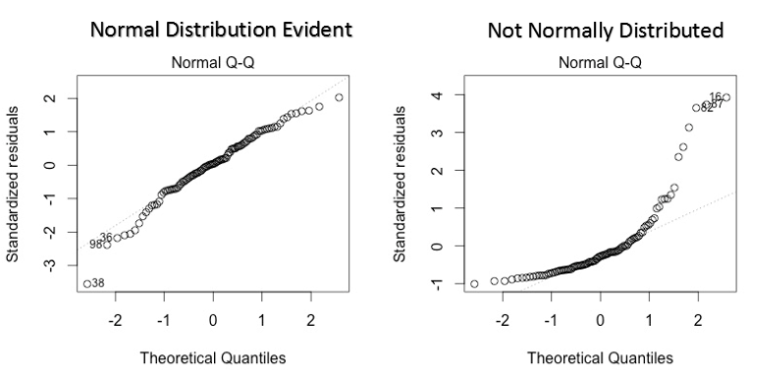
Below plots show that model does not fits the data well as pattern is evident in the plot:



# QQ PLOT

A QQ or quantile-quantile is a scatter plot which helps us validate the assumption of normal distribution in a data set. Using this plot, we can infer if the data comes from a normal distribution. If yes, the plot would show fairly straight line. Absence of normality in the errors can be seen with deviation in the straight line.

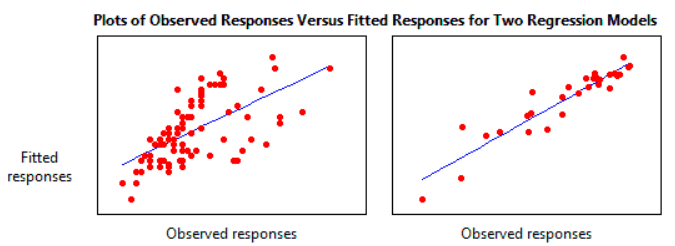
If you are wondering what is a ‘quantile’, here’s a simple definition: Think of quantiles as points in your data below which a certain proportion of data falls. Quantile is often referred to as percentiles. For example: when we say the value of 50th percentile is 120, it means half of the data lies below 120.



Solution: If the errors are not normally distributed, non – linear transformation of the variables (response or predictors) can bring improvement in the model.

# REGRESSION PLOTS

Plotting fitted values by observed values graphically illustrates different R-squared values for regression models.



The regression model on the left accounts for 38.0% of the variance while the one on the right accounts for 87.4%. The more variance that is accounted for by the regression model the closer the data points will fall to the fitted regression line. Theoretically, if a model could explain 100% of the variance, the fitted values would always equal the observed values and, therefore, all the data points would fall on the fitted regression line.

# MODEL SELECTION TECHNIQUE

We create all possible models using various learning algorithms applicable based on the business query we are trying to solve. All models are then cross validated using cross validation techniques and average scores for each model is calculated. Model with best cross validation is then chosen and tuned further.

# MODEL FIT EVALUATION

After you have fit a linear model using regression analysis, you need to determine how well the model fits the data.

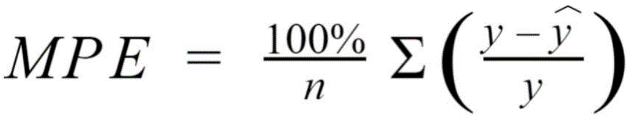
Linear regression calculates an equation that minimizes the distance between the fitted line and all of the data points. Technically, ordinary least squares (OLS) regression minimizes the sum of the squared residuals (Residual = Observed value - Fitted value).

In general, a model fits the data well if the differences between the observed values and the model's predicted values are small and unbiased. Goodness of fit for the model can be accessed using following model output parameters:

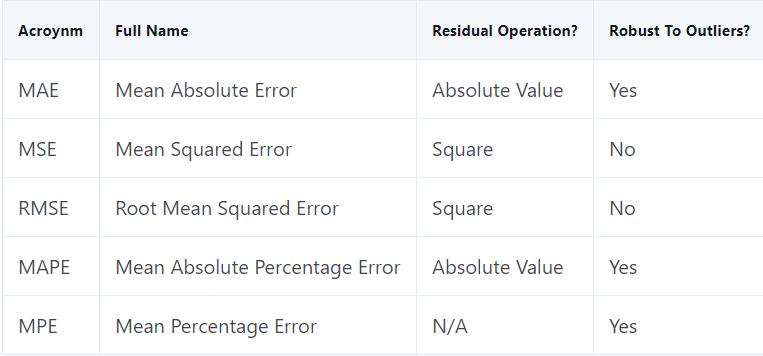
* AIC & BIC
* R squared value
* Adjusted R value
* Predicted R value
* RMSE
* Residual plot
* Bias – Variance

# MEAN PERCENTAGE ERROR

The mean percentage error (MPE) equation is given by below equation:



Since positive and negative errors will cancel out, we cannot make any statements about how well the model predictions perform overall. However, if there are more negative or positive errors, this bias will show up in the MPE. Unlike MAE and MAPE, MPE is useful to us because it allows us to see if our model systematically underestimates (more negative error) or overestimates (positive error).



Important points to remember:

1. Best line of fit is evaluated using cost function or Sum of squared error (SSE). The line with least SSE is termed as best fit line. Hence, we try to find features coefficients in such a way that YP ~ Y i.e. we try to minimize error / cost associated with the line.
2. We take squared difference of the error terms (loss) so the we get a convex function and we can find its global minima point.

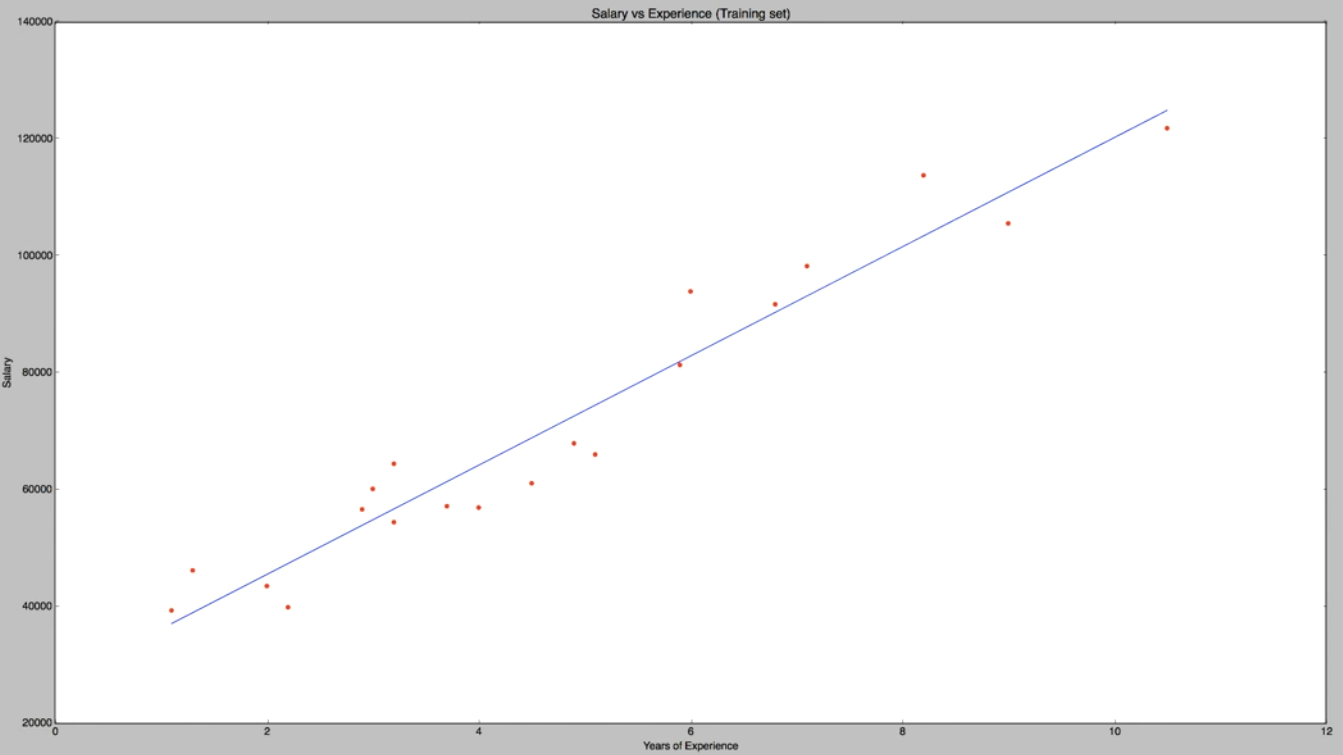
# STACKING REGRESSORS

<http://rasbt.github.io/mlxtend/user_guide/regressor/StackingRegressor/>

# REGRESSION MODEL OUTPUTS

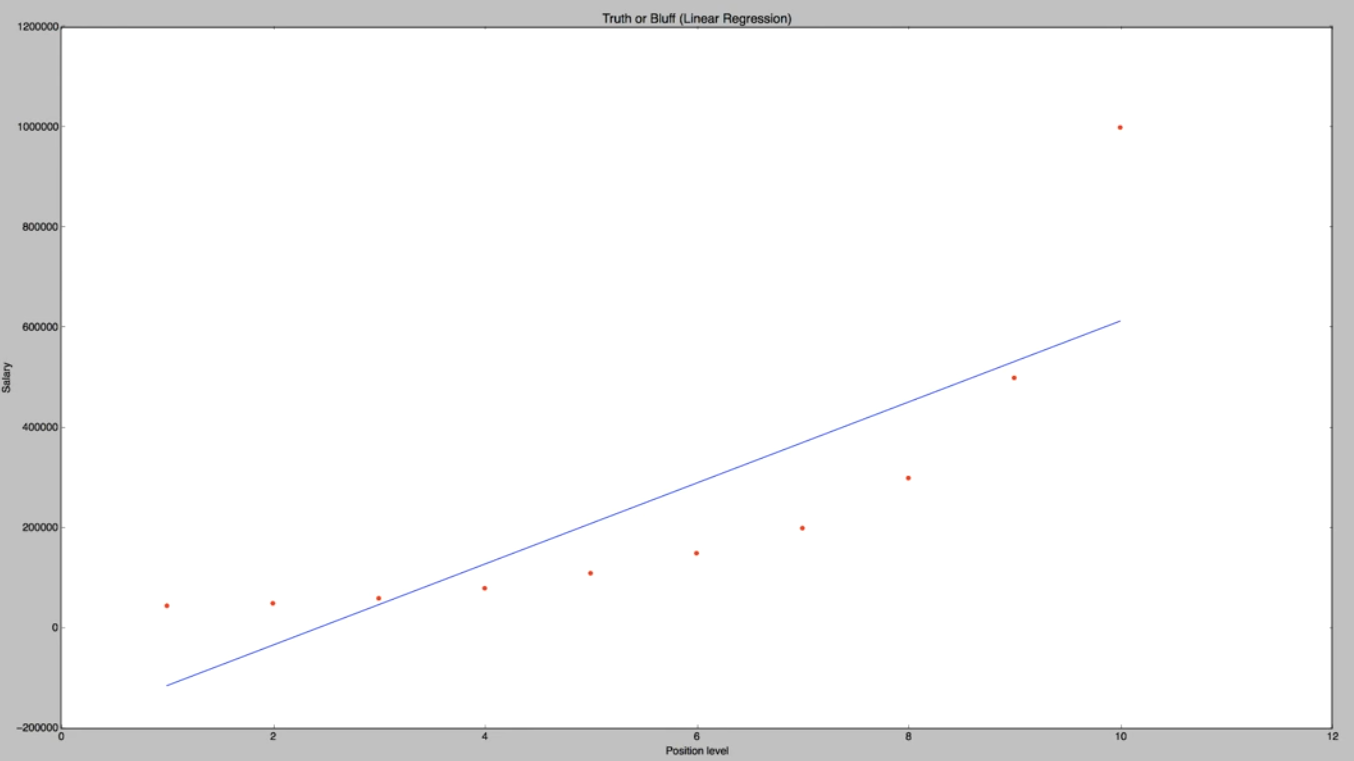
Below are the lines of fits for various regression models in 1D.

## LINEAR REGRESSION



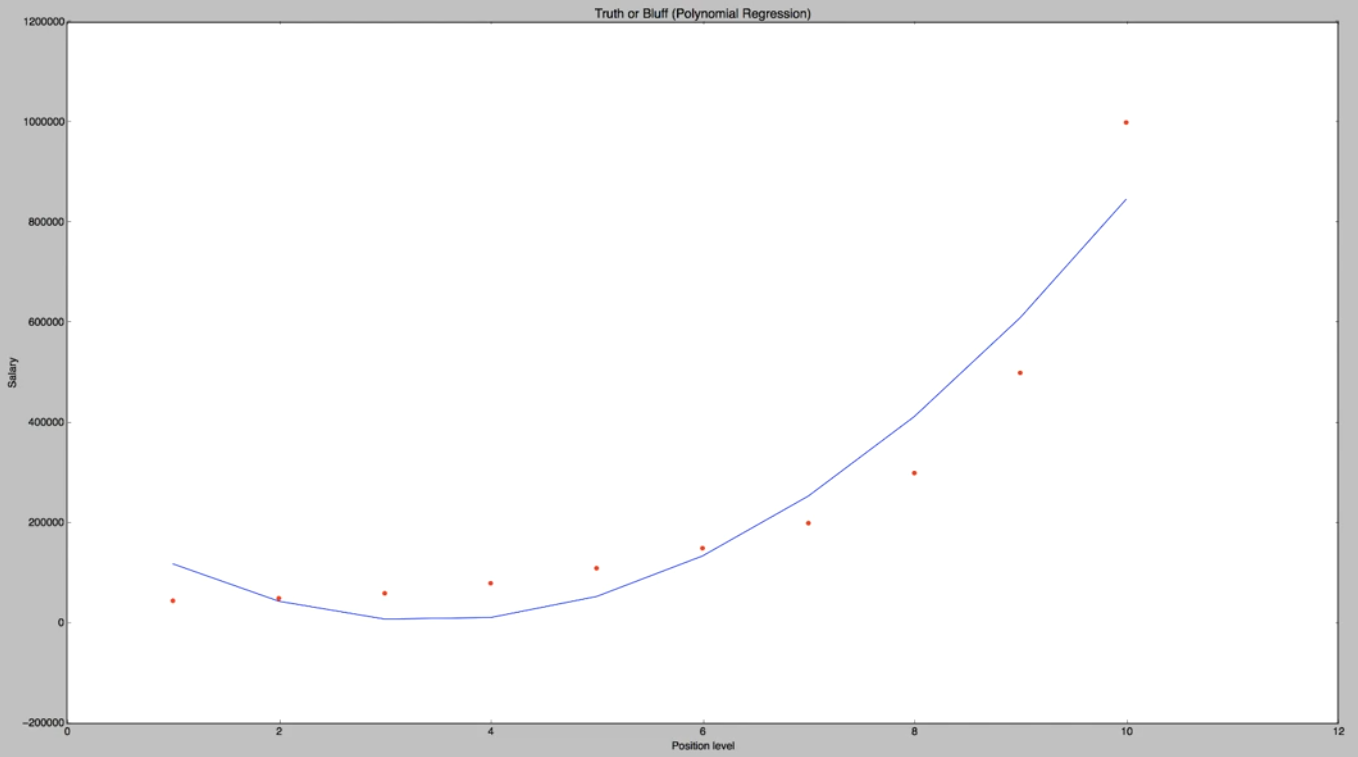
## LINEAR REGRESSION FOR NON-LINEAR PROBLEM

Outlier should be treated for linear regression if regularization is not used.



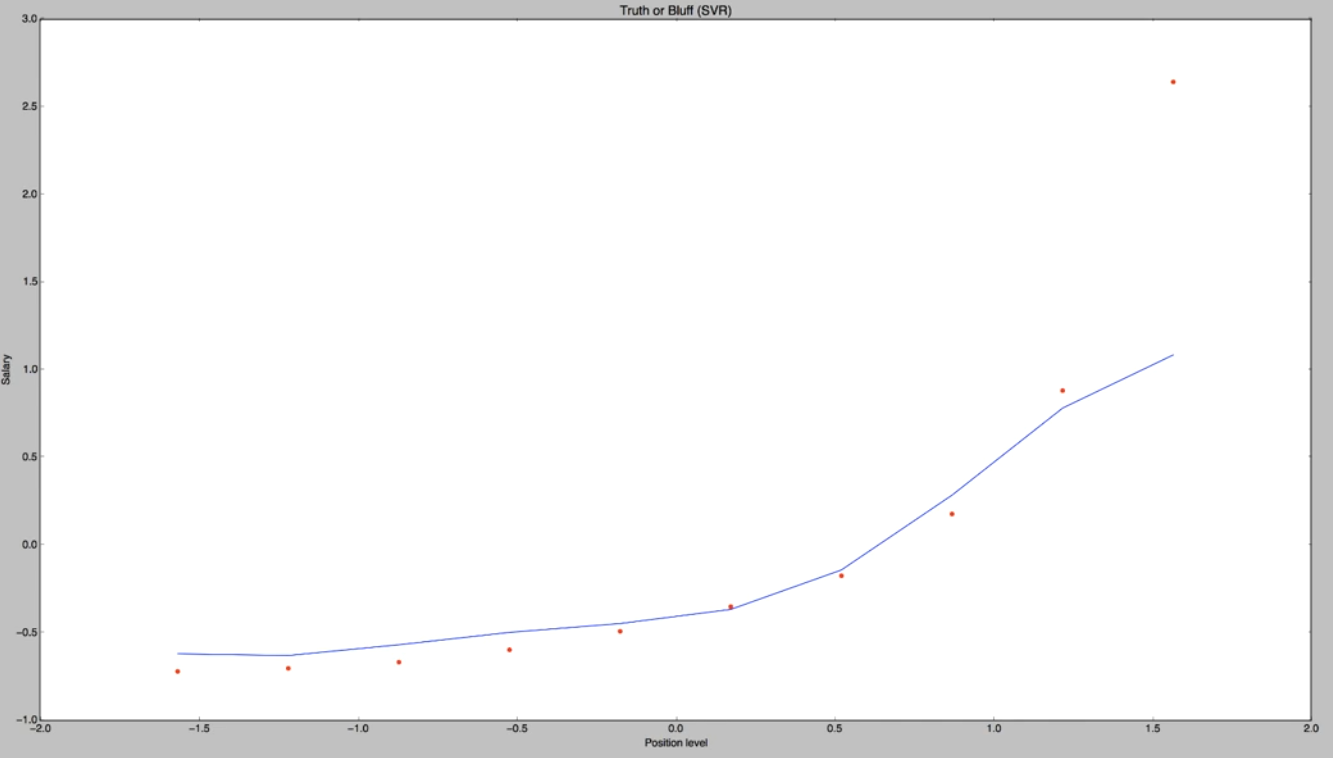
## POLYNOMIAL REGRESSION

Outlier has effect on prediction.



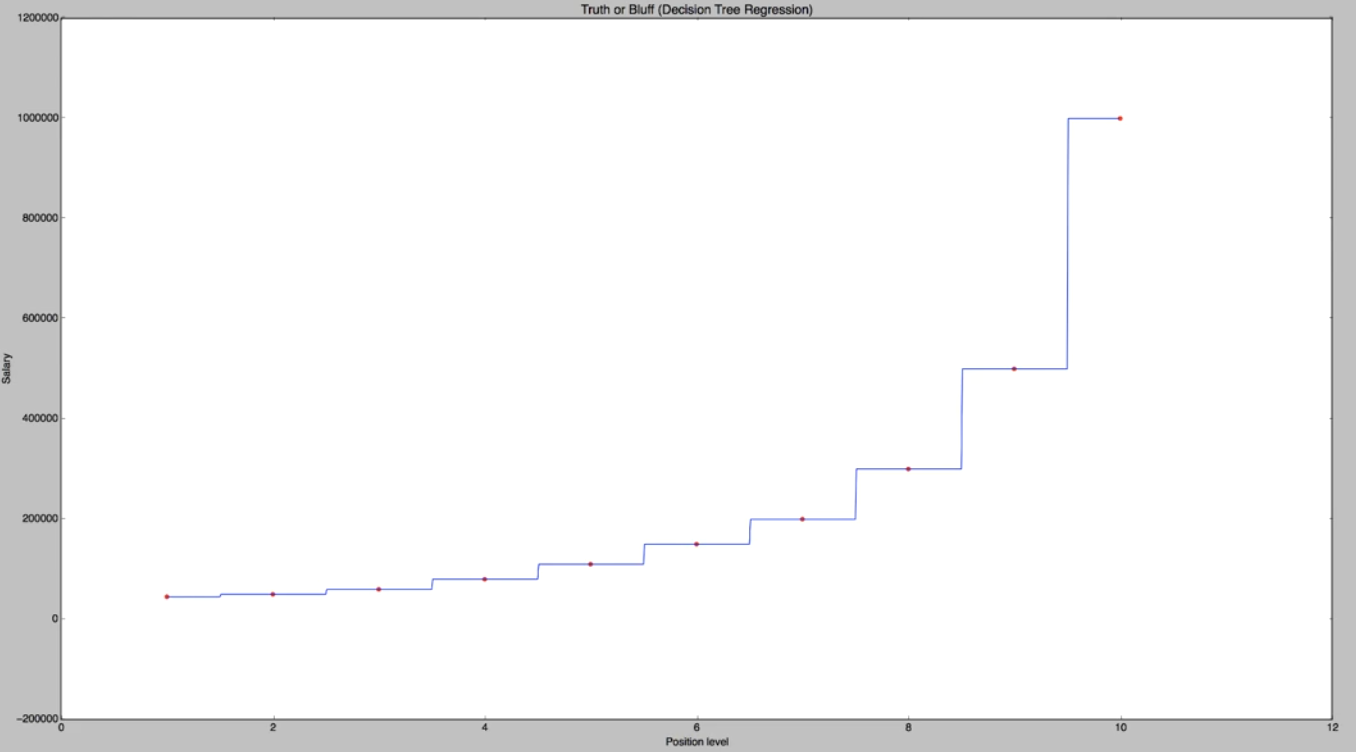
## SUPPORT VECTOR REGRESSION

As we can see below, SVR model does not consider outliers in the dataset for prediction as penalty parameters are included in the algorithm by default.



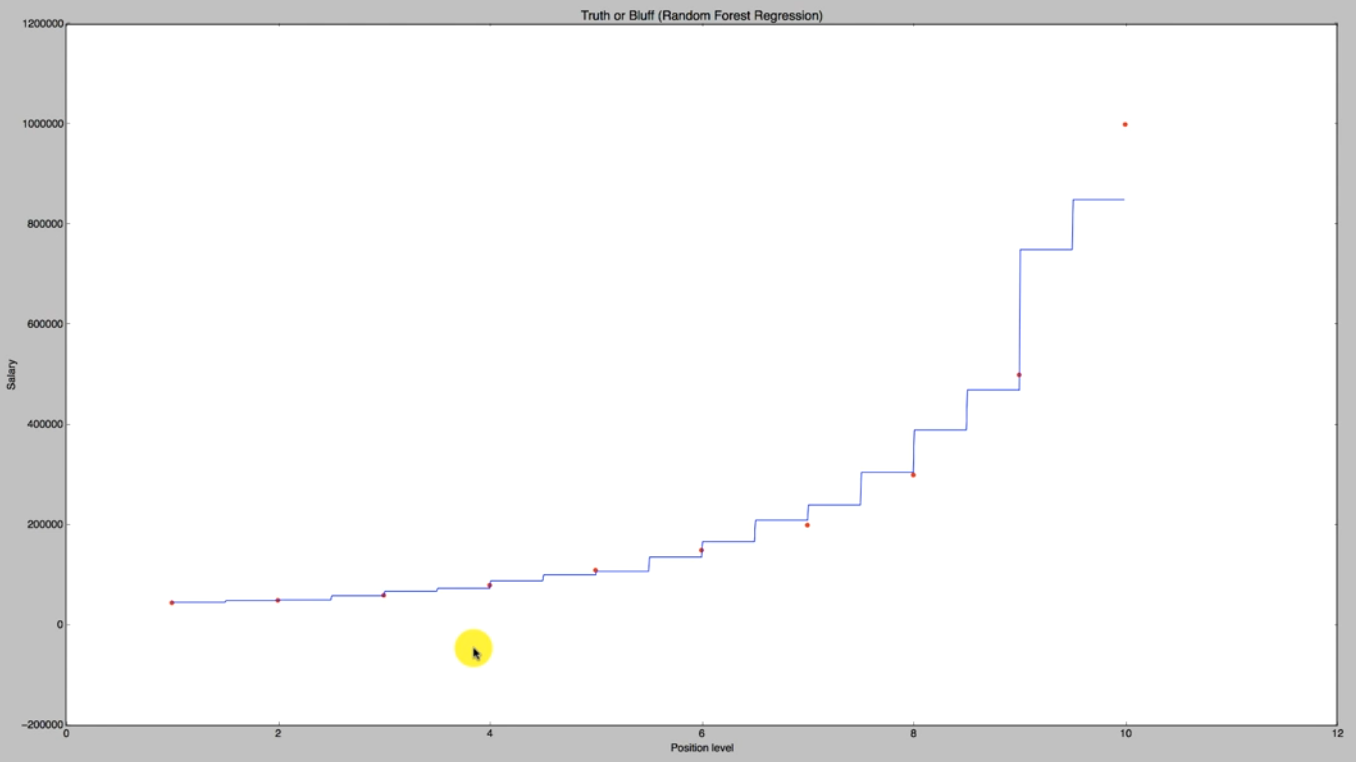
## DECISION TREE

As shown below, decision tree does not have a penalty term by default hence outlier is considered.



## RANDOM FOREST

We can include penalty parameter and hence outlier will not the predictions. Since many random trees are created and then mean of the trees are calculated, we can see multiple steps within an interval.



# REFERENCES

Sklearn documentation:

<https://scikit-learn.org/stable/>

Statsmodels documentation

<https://www.statsmodels.org/stable/py-modindex.html>

Difference between parameter and hyper-parameter

<https://machinelearningmastery.com/difference-between-a-parameter-and-a-hyperparameter/>

Overfitting and underfitting in ML

<https://machinelearningmastery.com/overfitting-and-underfitting-with-machine-learning-algorithms/>

Model Training parameters

<https://docs.aws.amazon.com/machine-learning/latest/dg/training-parameters.html>

Building a machine learning application

<https://docs.aws.amazon.com/machine-learning/latest/dg/building-machine-learning.html>

Feature Selection

<https://www.analyticsvidhya.com/blog/2016/12/introduction-to-feature-selection-methods-with-an-example-or-how-to-select-the-right-variables/>

<https://medium.com/@indreshbhattacharyya/feature-selection-categorical-feature-selection-boruta-light-gbm-chi-square-bf47e94e2558>

Type of Regressions

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

Ridge and Lasso Regression in Python

<https://www.analyticsvidhya.com/blog/2016/01/complete-tutorial-ridge-lasso-regression-python/>

Feature selection with bourta

<https://www.kaggle.com/rsmits/feature-selection-with-boruta>

Multicollinearity and resolution

<https://www.statisticssolutions.com/multicollinearity/>

<http://statisticsbyjim.com/regression/multicollinearity-in-regression-analysis/>

<http://blog.minitab.com/blog/understanding-statistics/handling-multicollinearity-in-regression-analysis>

Controlling BIAS and Variance

<https://towardsdatascience.com/balancing-bias-and-variance-to-control-errors-in-machine-learning-16ced95724db>

REGRESSION METRICS

<https://towardsdatascience.com/how-to-select-the-right-evaluation-metric-for-machine-learning-models-part-1-regrression-metrics-3606e25beae0>

Anomaly detection in Python

<https://www.datascience.com/blog/python-anomaly-detection>

Residual plot

<http://blog.minitab.com/blog/adventures-in-statistics-2/why-you-need-to-check-your-residual-plots-for-regression-analysis>

<https://www.kaggle.com/ekami66/detailed-exploratory-data-analysis-with-python>

Polynomial Regression

<https://towardsdatascience.com/polynomial-regression-bbe8b9d97491>

<https://medium.com/coinmonks/polynomial-regression-11bec9262d64>

Treating categorical variables

<https://www.analyticsvidhya.com/blog/2015/11/easy-methods-deal-categorical-variables-predictive-modeling/>

Loss Methods for Regression

<https://heartbeat.fritz.ai/5-regression-loss-functions-all-machine-learners-should-know-4fb140e9d4b0>