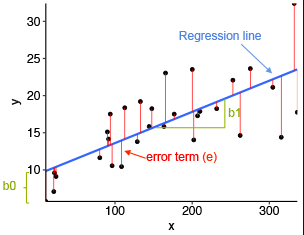
1. What is Regression?

Regression in statistics is the process of predicting a Label (or Dependent Variable) based on the features (Independent Variables). or estimating the relationship between the variables.

1. What is Linear Regression?

Linear Regression establishes a relationship between **dependent variable (Y)** and one or more **independent variables (X)** using a **best fit straight line** (also known as regression line). In this technique, the dependent variable is continuous, independent variable(s) can be [continuous or discrete](https://en.wikipedia.org/wiki/Continuous_and_discrete_variables), and nature of regression line is linear.

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



1. When to use Linear Regression? Explain the equation of a straight line.

Linear regression assumes that the relationship between your input and output is linear. So when linear relationship is there between depeandant and independatnt variables we should use linear regression.

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

1. What kind of plots will you use to show case the relationship amongst the columns?

Scatter plot – to find regression line

Heat map – to find correlation between variables

Scatter and surface plots – for multiple linear regression

1. How is the best fit line chosen?

Line of best fit is one of the most important outputs of [regression](https://www.investopedia.com/terms/r/regression.asp) analysis.

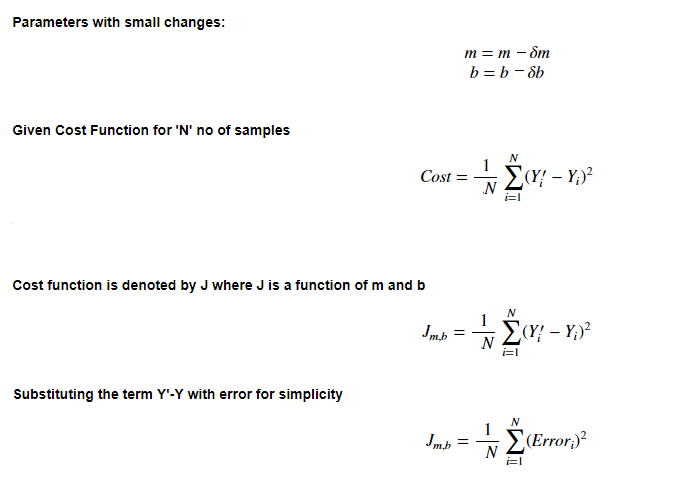
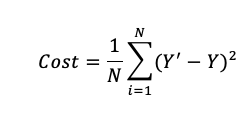
use the least squares method to arrive at the geometric equation for the line.

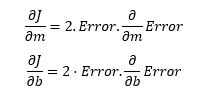
The vertical distance each point is above or below the line has been added to the diagram. These distances are called deviations or errors – they are symbolised as d d dn , ,..., 1 2 . When drawing in a regression line, the aim is to make the line fit the points as closely as possible. We do this by making the total of the squares of the deviations as small as possible, i.e. we minimise 2 i ∑d . If a line of best fit is found using this principle, it is called the least-squares regression line.

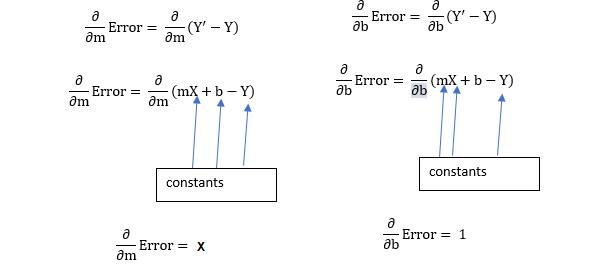
1. What is gradient descent, and why is it used? Explain the maths.

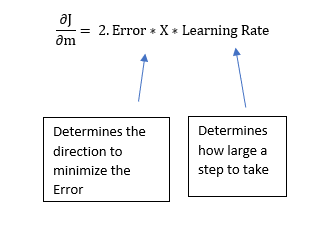
Gradient descent is used to minimize a cost function J(W) parameterized by a model parameters W. The gradient (or derivative) tells us the incline or slope of the cost function. Hence, to minimize the cost function, we move in the direction opposite to the gradient. Initialize the weights W randomly

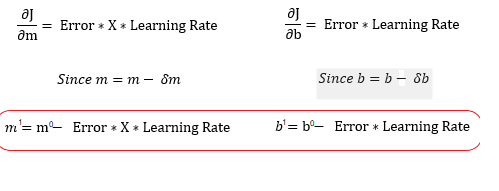
A **Cost Function/Loss Function** evaluates the performance of our Machine Learning Algorithm.

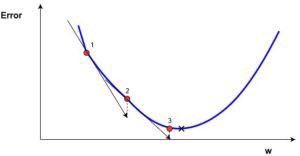
















1. What are residuals?

A residual is a measure of how well a line fits an individual data point.

**Residual = Observed value – predicted value  
e = y – ŷ**

Features of Residuals

Now that we have seen an example, there are a few features of residuals to note:

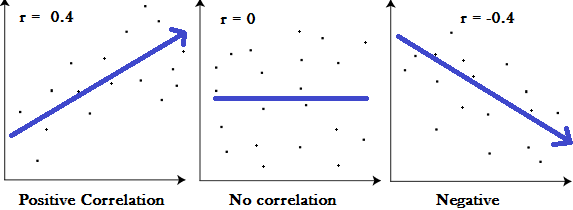
* Residuals are positive for points that fall above the regression line.
* Residuals are negative for points that fall below the regression line.
* Residuals are zero for points that fall exactly along the regression line.
* The greater the absolute value of the residual, the further that the point lies from the regression line.
* The sum of all of the residuals should be zero. In practice sometimes this sum is not exactly zero. The reason for this discrepancy is that roundoff errors can accumulate.

1. What is correlation?

**Correlation** is a term that is a measure of the strength of a linear relationship between two quantitative variables (e.g., height, weight).

Positive correlation is a relationship between two variables in which both variables move in the same direction. This is when one variable increases while the other increases and visa versa.

Whilst negative correlation is a relationship where one variable increases as the other decreases, and vice versa.



## Pearson’s Product-Moment Correlation

The most common measure of correlation is Pearson’s product-moment correlation, which is commonly referred to simply as the correlation, the [*correlation coefficient*](https://www.displayr.com/how-to-create-a-correlation-matrix-in-displayr/), or just the letter r (always written in italics). The correlation coefficient r measures the strength and direction of a linear relationship, for instance:

* 1 indicates a perfect positive correlation.
* -1 indicates a perfect negative correlation.
* 0 indicates that there is no relationship between the different variables.
* **sample correlation coefficient**is defined by the following formula, where *sx* and *sy* are the sample standard deviations, and *sxy* is the sample covariance.
*       s
  rxy =--xy
       sxsy
  

1. What is multicollinearity?

**Multicollinearity** is a state of very high intercorrelations or inter-associations among the independent variables. It is therefore a type of disturbance in the data, and if present in the data the statistical inferences made about the data may not be reliable.

1. How to detect multicollinearity?

An easy way to detect multicollinearity is to calculate [correlation coefficients](https://www.statisticshowto.com/probability-and-statistics/correlation-coefficient-formula/) for all pairs of predictor variables. If the correlation coefficient, r, is exactly +1 or -1, this is called perfect multicollinearity. If r is close to or exactly -1 or +1, one of the variables should be removed from the model if at all possible.

Using Variance Inflation Factor: Regression of one X variable against other X variables.

VIF=1(1−𝑅𝑠𝑞𝑢𝑎𝑟𝑒𝑑)1(1−Rsquared)

The VIF factor, if greater than 10 shows extreme correlation between the variables and then we need to take care of the correlation.

There is also another final statistical tool that is great for analyzing the variance between features. It is ANOVA which stands for Analysis of Variances. Generally, the higher the variance between the variables, the less likely that they are related (or correlated).

### **Steps for Implementing VIF**

1. Run a multiple regression.
2. Calculate the VIF factors.
3. Inspect the factors for each predictor variable, if the VIF is between 5-10, multicolinearity is likely present and you should consider dropping the variable.

<https://etav.github.io/python/vif_factor_python.html>

1. What are the remedies for multicollinearity?

**The potential solutions include the following:**

1. Remove some of the highly correlated independent variables.
2. Linearly combine the independent variables, such as adding them together.
3. Perform an analysis designed for highly correlated variables, such as principal components analysis or partial least squares regression.
4. What is the R-Squared Statistics?

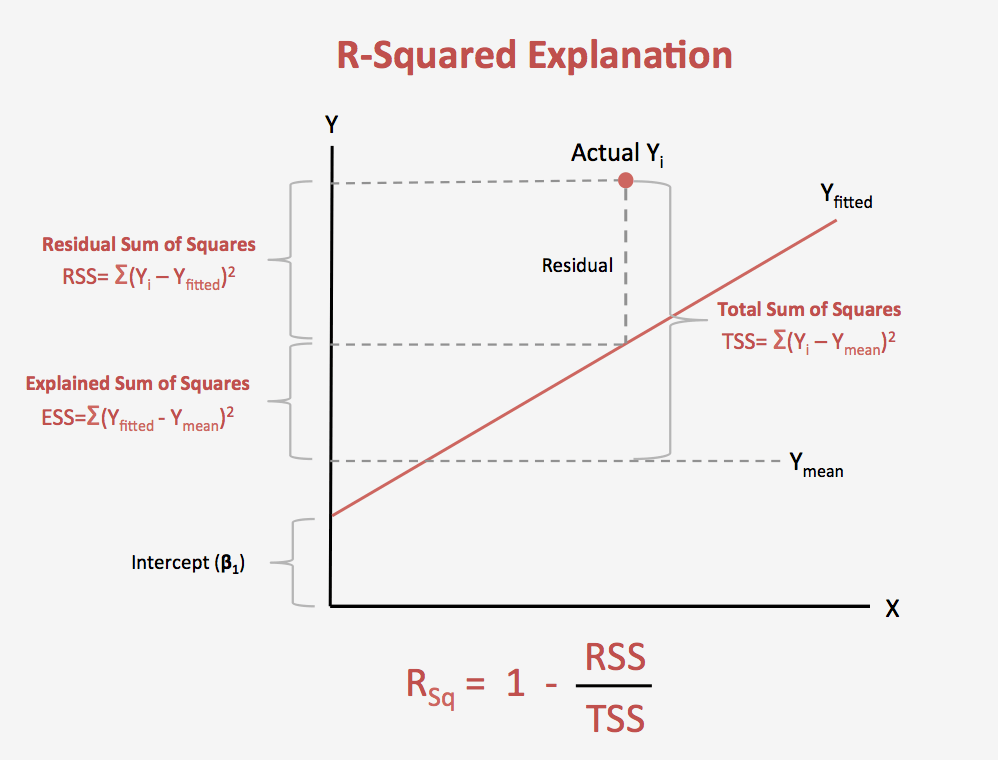
R-squared is a statistical measure of how close the data are to the fitted regression line. It is also known as the coefficient of determination, or the coefficient of multiple determination for multiple regression.

R-squared = Explained variation / Total variation

R-squared is always between 0 and 100%:

* 0% indicates that the model explains none of the variability of the response data around its mean.
* 100% indicates that the model explains all the variability of the response data around its mean.

In general, the higher the R-squared, the better the model fits your data.

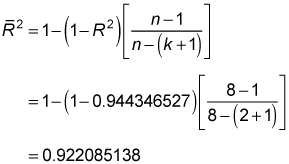


1. What is an adjusted R-Squared Statistics?

R2 shows how well terms (data points) fit a curve or line. Adjusted R2 also indicates how well terms fit a curve or line, but adjusts for the number of terms in a model. If you add more and more **useless** [variables](https://www.statisticshowto.com/probability-and-statistics/types-of-variables/)to a model, adjusted r-squared will decrease. If you add more **useful** variables, adjusted r-squared will increase.  
Adjusted R2 will always be less than or equal to R2.

You only need R2 when working with[**samples**](https://www.statisticshowto.com/sample/). In other words, R2 isn’t necessary when you have data from an entire [population](https://www.statisticshowto.com/what-is-a-population/).

The **adjusted R**-**squared** is a modified version of **R**-**squared** 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



1. Why do we use adj R-squared?

R2 increases with every predictor added to a model. As R2 always increases and never decreases, it can appear to be a better fit with the more terms you add to the model. This can be completely misleading.

Similarly, if your model has too many terms and too many high-order polynomials you can run into the problem of over-fitting the data. When you over-fit data, a misleadingly high R2 value can lead to misleading projections.

The adjusted R2 will penalize you for adding independent variables (K in the equation) that do not fit the model.

1. Why adj R-squared decreases when we use incompetent variables?

“Adjusted R square” comes to help. Adjusted R-square penalizes you for adding variables which do not improve your existing model.

Hence, if you are building Linear regression on multiple variable, it is always suggested that you use Adjusted R-squared to judge goodness of model. In case you only have one input variable, R-square and Adjusted R squared would be exactly same.

1. How to interpret a Linear Regression model?
2. What is the difference between fit, fit\_transform and predict methods?

The **fit**() function calculates the values of these parameters. The **transform** function applies the values of the parameters on the actual data and gives the normalized value. The **fit\_transform**() function performs both **in the** same step. Note that the same value is got whether we perform in 2 steps or **in a** single step.

Hence, every sklearn's transform's fit() just calculates the parameters (e.g. μμ and σσ in case of [StandardScaler](http://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.StandardScaler.html)) and saves them as an internal objects state. Afterwards, you can call its transform() method to apply the transformation to a particular set of examples.

fit\_transform() joins these two steps and is used for the initial fitting of parameters on the training set xx, but it also returns a transformed x′x′. Internally, it just calls first fit() and then transform() on the same data.

scaler = StandardScaler()

scaler.fit(X\_train) # get the 2 parameters from data (\*\*μ and σ\*\*)

scaler.transform(X\_train) # apply scale with given parameters

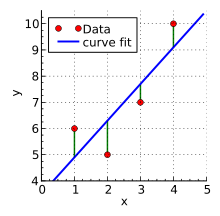
scaler.transform(X\_test) # apply scale with training parameters on the testing data

you can use fit\_transform(X\_train) for shortcut rather than fit(X\_train) => transform(X\_train)

1. How do you plot the least squared line?

*The* **goodness of fit** *of a line* yˆ=mx+by^=mx+b *to a set of* *n* *pairs* (x,y)(x,y) *of numbers in a sample is the sum of the squared errors*

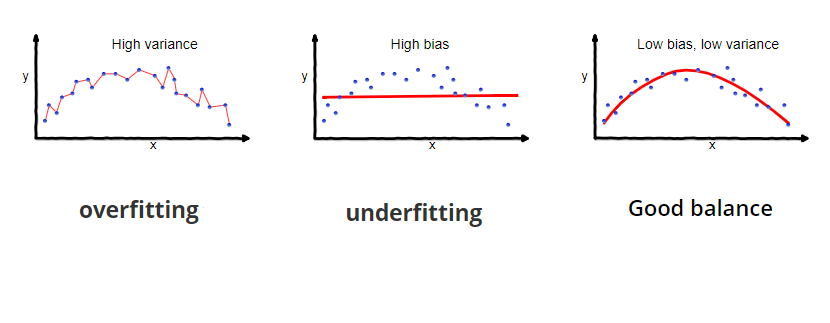
Σ(y−yˆ)2



1. What are Bias and Variance? What is Bias Variance Trade-off?

Bias is the difference between the average prediction of our model and the correct value which we are trying to predict. Model with high bias pays very little attention to the training data and oversimplifies the model. It always leads to high error on training and test data.

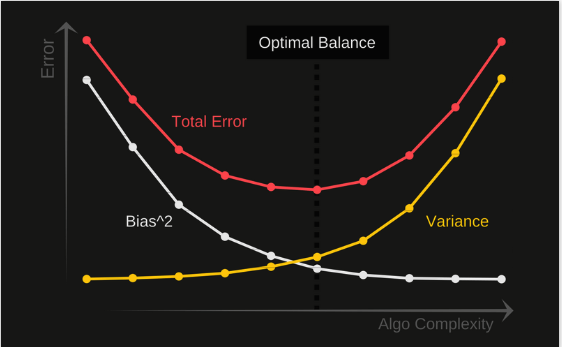
Variance is the variability of model prediction for a given data point or a value which tells us spread of our data. Model with high variance pays a lot of attention to training data and does not generalize on the data which it hasn’t seen before. As a result, such models perform very well on training data but has high error rates on test data.



To build a good model, we need to find a good balance between bias and variance such that it minimizes the total error.

An optimal balance of bias and variance would never overfit or underfit the model.

https://miro.medium.com/max/441/1*SKHGhoGKnBh_GPGHI2Ktvw.png



Examples of **low-bias** machine learning algorithms include: Decision Trees, k-Nearest Neighbors and [Support Vector Machines](https://machinelearningmastery.com/support-vector-machines-for-machine-learning/).

Examples of **high-bias** machine learning algorithms include: Linear Regression, Linear Discriminant Analysis and Logistic Regression.

Variance is the amount that the estimate of the target function will change if different training data was used.

Examples of **low-variance** machine learning algorithms include: Linear Regression, Linear Discriminant Analysis and Logistic Regression.

Examples of **high-variance** machine learning algorithms include: Decision Trees, k-Nearest Neighbors and Support Vector Machines.

* **Linear** machine learning algorithms often have a high bias but a low variance.
* **Nonlinear** machine learning algorithms often have a low bias but a high variance.

### Linear Algorithms

* Linear Regression
* Logistic Regression
* Linear Discriminant Analysis

### Nonlinear Algorithms

* Classification and Regression Trees
* Naive Bayes
* k-Nearest Neighbors
* Learning Vector Quantization
* Support Vector Machines

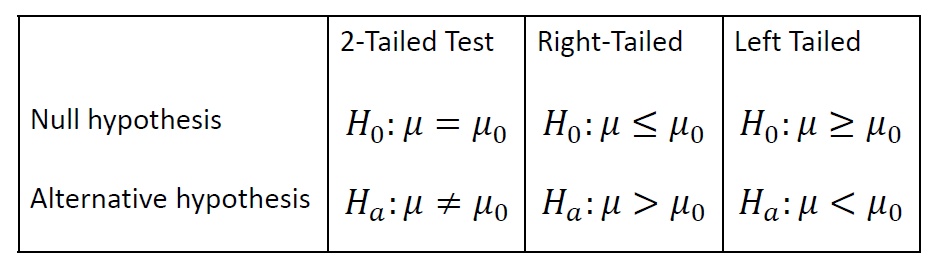
### Ensemble Algorithms

* Bagging and Random Forest
* Boosting and AdaBoost

1. What is the null and alternate hypothesis?

**Null Hypothesis:** The sample mean is equal to the proposed population mean

**Alternative Hypothesis:** The sample mean is not equal to the proposed population mean



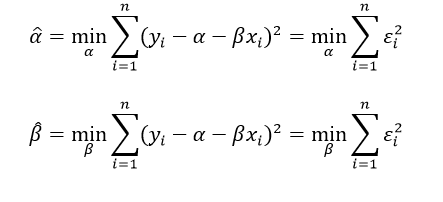
1. What is multiple linear regression?

If we have multiple features and create a model to see the relationship between those features and the label column. This is called **Multiple Linear Regression**.

𝑦=𝛽0+𝛽1𝑥1+...+𝛽𝑛𝑥𝑛

1. What is the OLS method? Derive the formulae used in the OLS method.

**ordinary least squares** (**OLS**) is a type of [linear least squares](https://en.wikipedia.org/wiki/Linear_least_squares) method for estimating the unknown [parameters](https://en.wikipedia.org/wiki/Statistical_parameter) in a [linear regression](https://en.wikipedia.org/wiki/Linear_regression) model. OLS chooses the parameters of a [linear function](https://en.wikipedia.org/wiki/Linear_function) of a set of [explanatory variables](https://en.wikipedia.org/wiki/Explanatory_variable) by the principle of [least squares](https://en.wikipedia.org/wiki/Least_squares): minimizing the sum of the squares of the differences between the observed [dependent variable](https://en.wikipedia.org/wiki/Dependent_variable) (values of the variable being observed) in the given [dataset](https://en.wikipedia.org/wiki/Dataset) and those predicted by the linear function.



1. What is the p-value? How does it help in feature selection?

It's a measure of how extreme an observed value is under the assumed null hypothesis: the smaller it is, the more extreme the observation. We can define *p*-value as the **smallest** significance level at which the null hypothesis would be rejected.

*Alpha value is nothing but****a threshold p-value,***w*hich the group conducting the test/experiment decides upon before conducting a test of similarity or significance ( Z-test or a T-test).*

Removal of different features from the dataset will have different effects on the p-value for the dataset. We can remove different features and measure the p-value in each case. These measured p-values can be used to decide whether to keep a feature or not.

p values **don’t help in feature selection**as p-values fail to detect important features!

Use **RFECV (R**ecursive **F**eature **E**limination with **C**ross-**V**alidation)

RFECV-as its title suggests-recursively removes features, builds a model using the remaining attributes and calculates model accuracy.

1. **from** sklearn.datasets **import** make\_friedman1
2. **from** sklearn.feature\_selection **import** RFECV
3. **from** sklearn.svm **import** SVR
4. X, y = make\_friedman1(n\_samples=50, n\_features=10, random\_state=0)
5. estimator = SVR(kernel="linear")
6. selector = RFECV(estimator, step=1, cv=5)
7. selector = selector.fit(X, y)
8. selector.support\_
9. selector.ranking\_

Output:

1. array([1, 1, 1, 1, 1, 6, 4, 3, 2, 5])

It means that if there were 10 features, their ranks are 1, 1, 1, 1, 1, 6, 4, 3, 2, 5. Take the top 3 features, i.e, features with rank 1 (first 5 features) and with rank 2 (9th feature) and rank 3 (8th feature)

<https://towardsdatascience.com/feature-selection-correlation-and-p-value-da8921bfb3cf>

1. How to handle categorical values in the data?

Use Category Encoders to improve model performance when you have nominal or ordinal data that may provide value.

For nominal columns try OneHot, Hashing, LeaveOneOut, and Target encoding. Avoid OneHot for high cardinality columns and decision tree-based algorithms.

For ordinal columns try Ordinal (Integer), Binary, OneHot, LeaveOneOut, and Target. Helmert, Sum, BackwardDifference and Polynomial are less likely to be helpful, but if you have time or theoretic reason you might want to try them.

For regression tasks, Target and LeaveOneOut probably won’t work well.

<https://towardsdatascience.com/smarter-ways-to-encode-categorical-data-for-machine-learning-part-1-of-3-6dca2f71b159>

1. What is regularization, and why do we need it?

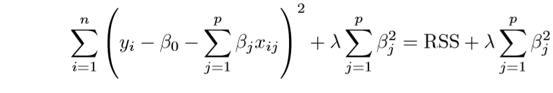
Regularizations **are** techniques used to reduce the error by fitting a function appropriately on the given training set and avoid overfitting

When we use regression models to train some data, there is a good chance that the model will overfit the given training data set. Regularization helps sort this overfitting problem by restricting the degrees of freedom of a given equation i.e. simply reducing the number of degrees of a polynomial function by reducing their corresponding weights.  
In a linear equation, we do not want huge weights/coefficients as a small change in weight can make a large difference for the dependent variable (Y). So, regularization constraints the weights of such features to avoid overfitting

1. Explain Ridge Regression.

Ridge regression penalizes the model based on the sum of squares of magnitude of the coefficients. The regularization term is given by regularization=𝜆∗∑|𝛽2𝑗|λ∗∑|βj2|

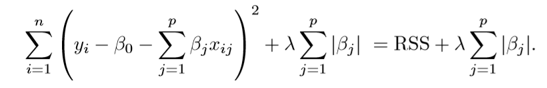
Where, λ is the shrinkage factor.and hence the formula for loss after regularization is:

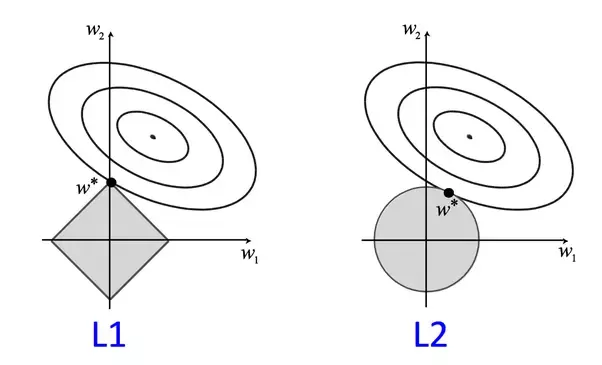


1. Explain Lasso Regression.

LASSO regression penalizes the model based on the sum of magnitude of the coefficients. The regularization term is given by regularization=𝜆∗∑|𝛽𝑗|λ∗∑|βj| Where, λ is the shrinkage factor.

and hence the formula for loss after regularization is:



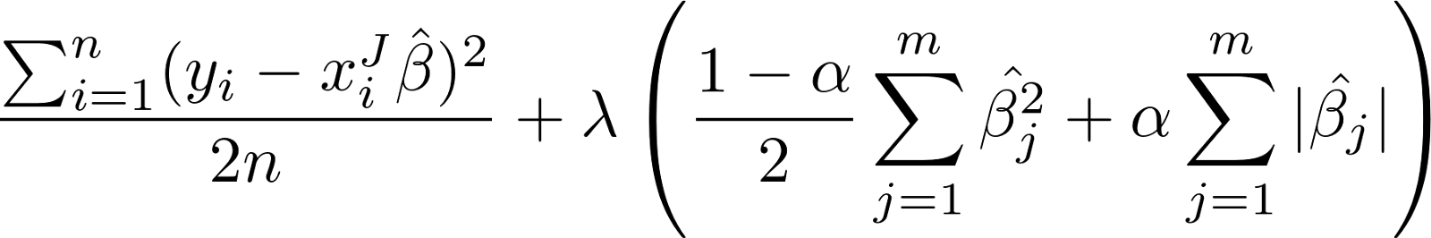


<https://codeburst.io/what-is-regularization-in-machine-learning-aed5a1c36590>

1. Explain Elastic Net.

According to the Hands-on Machine Learning book, elastic Net is a middle ground between Ridge Regression and Lasso Regression. The regularization term is a simple mix of both Ridge and Lasso’s regularization terms, and you can control the mix ratio α.

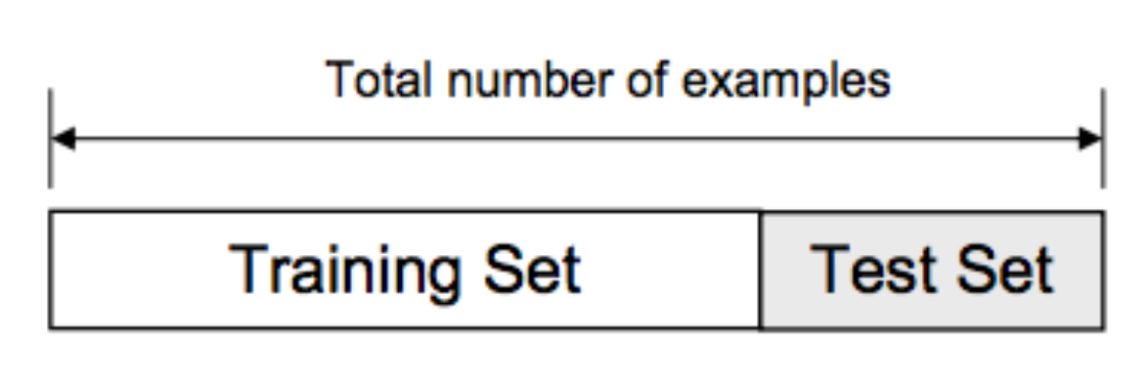
where α is the mixing parameter between ridge (α = 0) and lasso (α = 1).



In addition to setting and choosing a lambda value elastic net also allows us to tune the alpha parameter where 𝞪 = 0 corresponds to ridge and 𝞪 = 1 to lasso. Simply put, if you plug in 0 for alpha, the penalty function reduces to the L1 (ridge) term and if we set alpha to 1 we get the L2 (lasso) term. Therefore we can choose an alpha value between 0 and 1 to optimize the elastic net. Effectively this will shrink some coefficients and set some to 0 for sparse selection.

1. Why do we do a train test split?

**We split** into training data and **test** data. The training set contains a known output and the model learns on this data in order to be generalized to other data later on. **We** have the **test** dataset (or subset) in order to **test** our model's prediction on this subset



1. What is polynomial regression? When to use it?

Polynomial Regression can be defined as a mechanism to predict a *dependent variable* based on the polynomial relationship with the *independent variable*.

In the equation,

1. 𝑦=𝑎+𝑏𝑥+𝑐𝑥2+...+𝑛𝑥𝑛+...y=a+bx+cx2+...+nxn+...

the maximum power of 'x' is called the degree of the polynomial equation. For example, if the degree is 1, the equation becomes linear regression y = a+bx.

Many times we may face a requirement where we have to do a regression, but when we plot a graph between a dependent and independent variables, the graph doesn't turn out to be a linear one.

It means that the relationship between X and Y can't be described Linearly. Then comes the time to use the Polynomial Regression.

A polynomial term–a quadratic (squared) or cubic (cubed) term turns a linear regression model into a curve.

1. Explain the steps for GCP deployment.
   1. Go to <https://cloud.google.com/> and create an account if already haven’t created one. Then go to the console of your account.
   2. Go to IAM and admin(highlighted) and click manage resources
   3. Click CREATE PROJECT to create a new project for deployment.
   4. Once the project gets created, select App Engine and select Dashboard.
   5. Go to <https://dl.google.com/dl/cloudsdk/channels/rapid/GoogleCloudSDKInstaller.exe> to download the google cloud SDK to your machine.
   6. Click Start Tutorial on the screen and select Python app and click start.
   7. Check whether the correct project name is displayed and then click next.
   8. Create a file ‘app.yaml’ and put ‘runtime: python37’ in that file.
   9. Create a ‘requirements.txt’ file by opening the command prompt/anaconda prompt, navigate to the project folder and enter the command ‘pip freeze > requirements.txt’. It is recommended to use separate environments for different projects.
   10. Your python application file should be called ‘main.py’. It is a GCP specific requirement.
   11. Open command prompt window, navigate to the project folder and enter the command gcloud init to initialise the gcloud context.
   12. It asks you to select from the list of available projects.
   13. Once the project name is selected, enter the command gcloud app deploy app.yaml --project
   14. After executing the above command, GCP will ask you to enter the region for your application. Choose the appropriate one.
   15. GCP will ask for the services to be deployed. Enter ‘y’ to deploy the services.
   16. And then it will give you the link for your app.
2. What difficulties did you face in cloud deployment?
   1. Authorization errors wile logging in GCP
   2. Deployment errors because of versions in requirements file.

Interview questions:

<https://www.analyticsvidhya.com/blog/2017/07/30-questions-to-test-a-data-scientist-on-linear-regression/>