**Time complexity equation** works by inserting the size of the data set as an integer n, and returning the number of operations that need to be conducted by the computer before the function can finish.

Measurement of Efficiency–

Accessing - Searching - Inserting - Deleting

Modeled by an equation which takes in size of dataset(n) 🡨 (# of elements) and returns

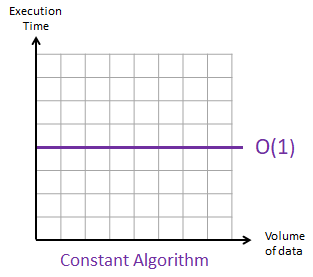
**number of operations** needed to be performed by the computer to complete that task.

↓

We can use this to see what a data structure is good at and what the data structure is bad at.

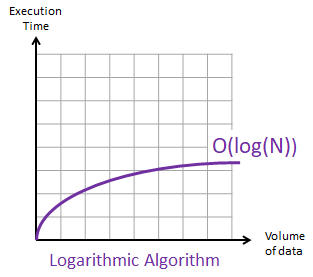
The absolute **best** a data structure can Score on each criteria is O(1):

* No matter what the size of your data set is, the task will be completed in a **single instruction**.



The next fastest type of time complexity equation is O(log n)

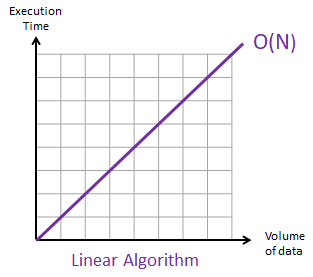
* Still provides fast completion time
* Gets more efficient as the size of the data set increases
* Slower than 0(1) faster than 0(N)
* The amount of data that you try and preform in the criteria listed above increases , the rate of change to complete the certain task decreases.



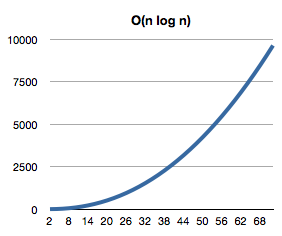
Example of 0(log(n)) is the binary search

The next in the series of types of time complexity is 0(N)

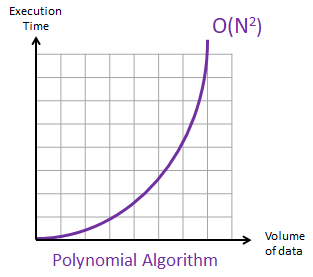
* The last of the decent equations
* For every element in the dataset so will the number of operations.

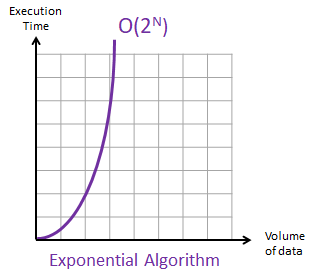


Another type of time complexity is O(n log n) , the slope increases as the volume of data increases ; making this relatively inefficient for larger datasets.

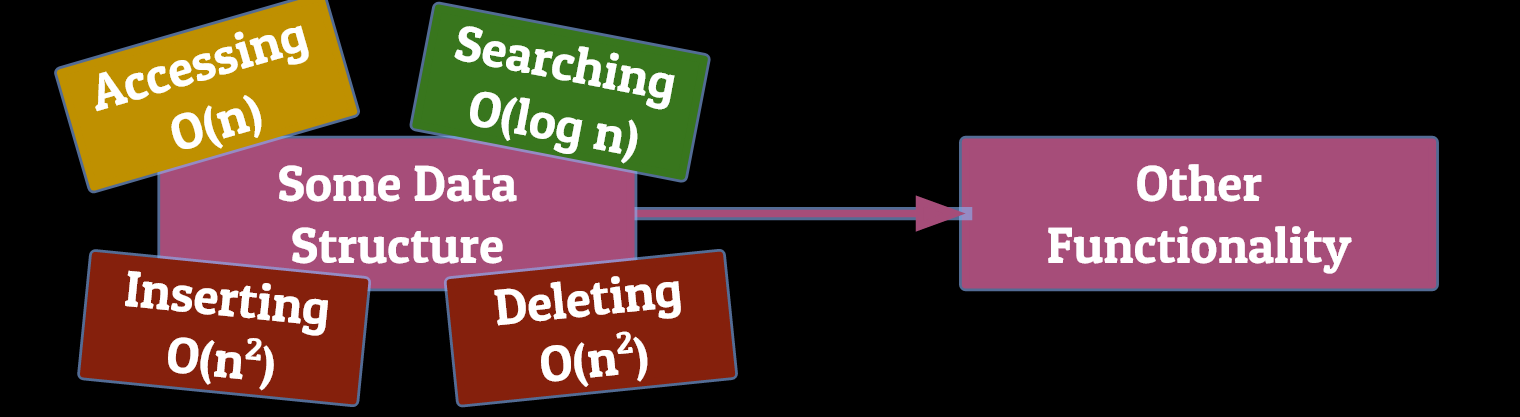


Same goes for the absolute rating for time complexity O(n^2) and O(2^n). Both are exponential in structure.





* Time Complexity Equations are NOT the only metric you should be using the gauge which data structure to use
  + Some provide other functionality that make them extremely useful

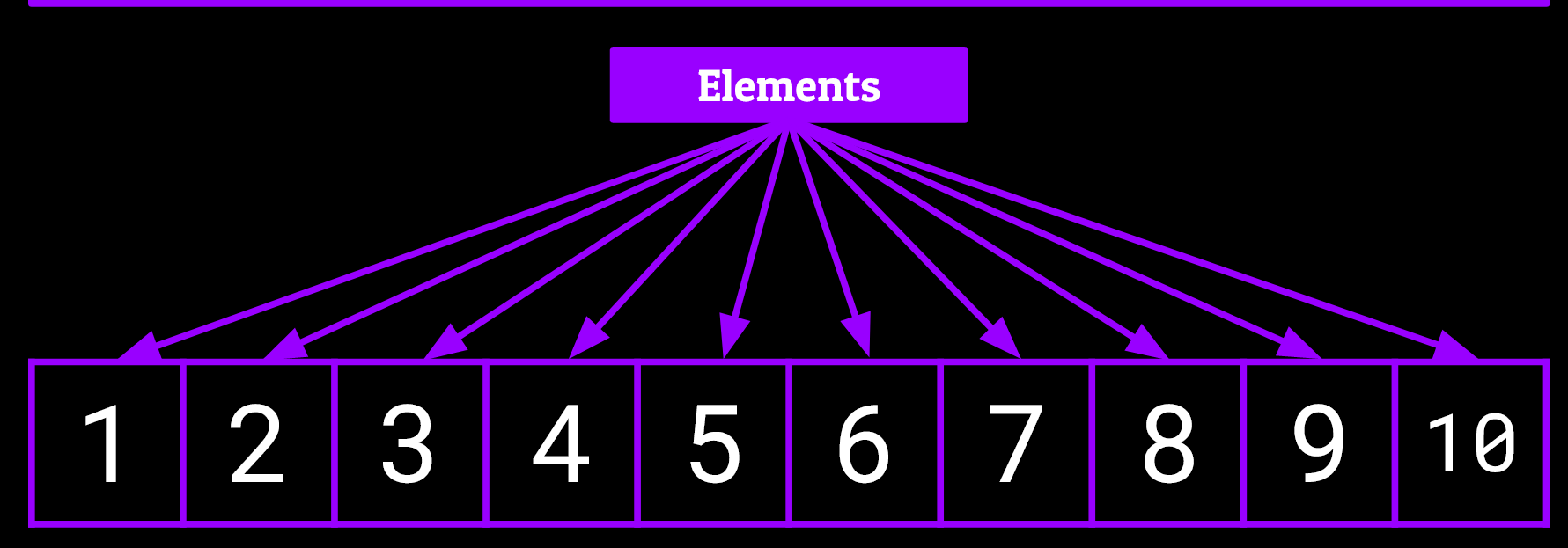


Data Structure

Array

* An array is fundamentally a list of similar values and can be used to store anything (usernames, high scores, prices). The data is stored in values of the same type (Integer, string, float).
* Every item in the list of data is referred to as an “element”

The collective total of elements is the array.



**↑**

**This concept of elements being what makes up an array is similar across all data structures.**

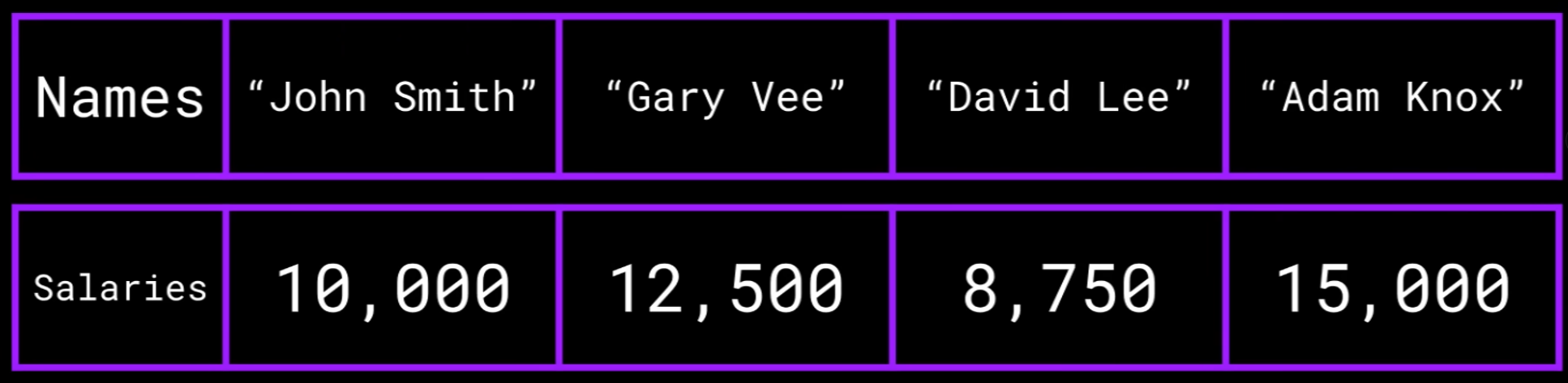
An array usually has 3 attributes associated with it.

* A Name
* A Type
* A Size

A name is simply a name for the array to reference with and interact with the dataset.

Parallel arrays

* 2 or more arrays which contain the same number of elements and have corresponding values in the same position.



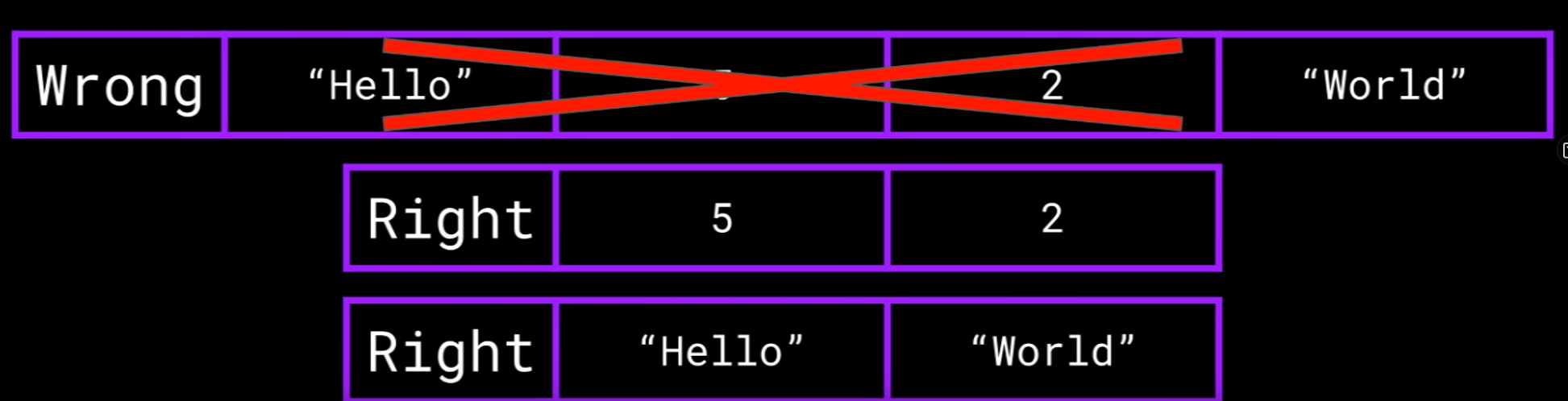
* Parallel arrays are extremely useful for storing differing types of data about the same entity.

Types:

An arrays type is simply what type of information is stored or will be stored within that array.

It MUST hold the same type of information.

Ex …



Size:

An array size is a set integer that is fixed upon the creation of the array.

* Represents the total amount of elements that are able to be stored within the array.
* The size cannot be changed.

**Machine Learning**:

Regulated Linear Models

Overfitting: When the dataset has been trained with too much information.

* + Poor input data gathered from your x-input functions results can skew the

training data.

↓

This leads to your model over generalizing information and has either a 100% accuracy in

the training model and is acting worse on validation accuracy.

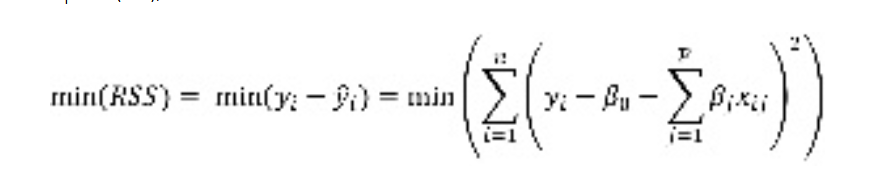
Solutions to Overfitting –

1. Reduce some features.
2. Regularization – eliminating features that are not useful or have as fun of an impact on target data.

Method 1: Lasso Regression (Strict)

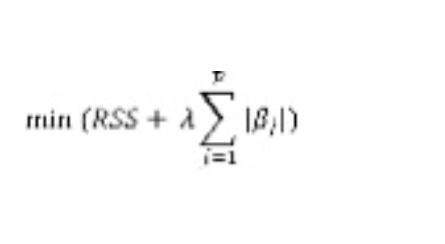
1.

Ordinary Least Squares regression choose the beta coefficient that minimizes the residual sum of squares, which is the difference between the observed Y and the estimated Ys.



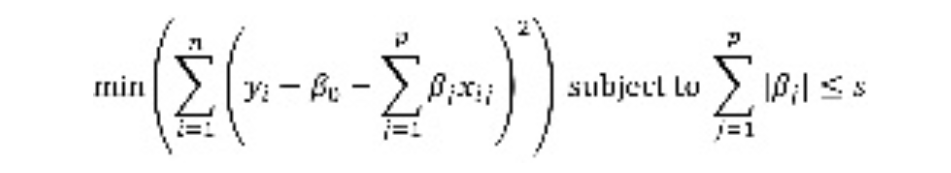
The Lasso is an extension of OLS which adds penalty to the RSS equal sum of the absolute values of the non-intercept beta coefficients multiplied by parameter λ (accuracy) that slows or accelerates the penalty (error). If λ is less than 1, it slows the penalty and if it is above 1 it accelerates the penalty.

2.



The LASSO can also be rewritten to be minimizing the RSS subject to the sum of the absolute values of the non-intercept beta coefficients being less than a constraint s. As s decreases toward 0, the beta coefficients shrink toward zero with the least associated beta coefficients decreasing all the way to 0 before the more strongly associated beta coefficients. As a result, numerous beta coefficients that are not strongly associated with the outcome are decreased to zero, which is equivalent to removing those variables from the model. In this way, the LASSO can be used as a variable selection method.

3.



In order to find the optimal λ, a range of λ's are tested, with the optimal λ chosen using cross-validation. Cross-validation involves:

* Separating the data into a training set and a test set
* Building the model in the training set
* Estimating the outcome in the test set using the model from the training set
* Calculating MSE in the test set

Method 2: Ridge Regression

Ridge [regression](https://www.mygreatlearning.com/blog/what-is-regression/) is a model tuning method that is used to analyses any data that suffers from multicollinearity **(** is a stats concept where serval independent variables in a model are correlated, Two variables are considered to be perfectly collinear if their correlation coefficient is +/- 1.0. Multicollinearity among independent variables will result in less reliable statistic inferences**)**. This method performs L2 regularization. When the issue of multicollinearity occurs, least-squares are unbiased, and variances are large, this results in predicted values being far away from the actual values.

**Min(||Y – X(theta)||^2 + λ||theta||^2)**

* **Theta = the weight of the feature in column**

Lambda is the penalty term. λ given here is denoted by an alpha parameter in the ridge function. So, by changing the values of alpha, we are controlling the penalty term. The higher the values of alpha, the bigger is the penalty and therefore the magnitude of coefficients is reduced.

* It shrinks the parameters. Therefore, it is used to prevent multicollinearity
* It reduces the model complexity by coefficient shrinkage
* Check out the free course on [regression analysis](https://www.mygreatlearning.com/academy/learn-for-free/courses/regression-analysis-with-excel-hands-on?gl-blog_id=20944).

**Ridge Regression Models**

* For any type of regression machine learning model, the usual regression equation forms the base which is written as:

***Y = XB + e***

Where Y is the dependent variable, X represents the independent variables, B is the regression coefficients to be estimated, and e represents the errors are residuals.

Once we add the lambda function to this equation, the variance that is not evaluated by the general model is considered. After the data is ready and identified to be part of L2 regularization, there are steps that one can undertake.

## ****Bias and variance trade-off****

Bias and variance trade-off is generally complicated when it comes to building ridge regression models on an actual dataset. However, following the general trend which one needs to remember is:

1. The bias increases as λ increases.
2. The variance decreases as λ increases.

The Ridge Regression is only supposed to be used at the end of the cost function when calculating the relationship between datasets.

Ridge regression is a method of estimating the coefficients of multiple-regression models in scenarios where the independent variables are highly correlated.

OR

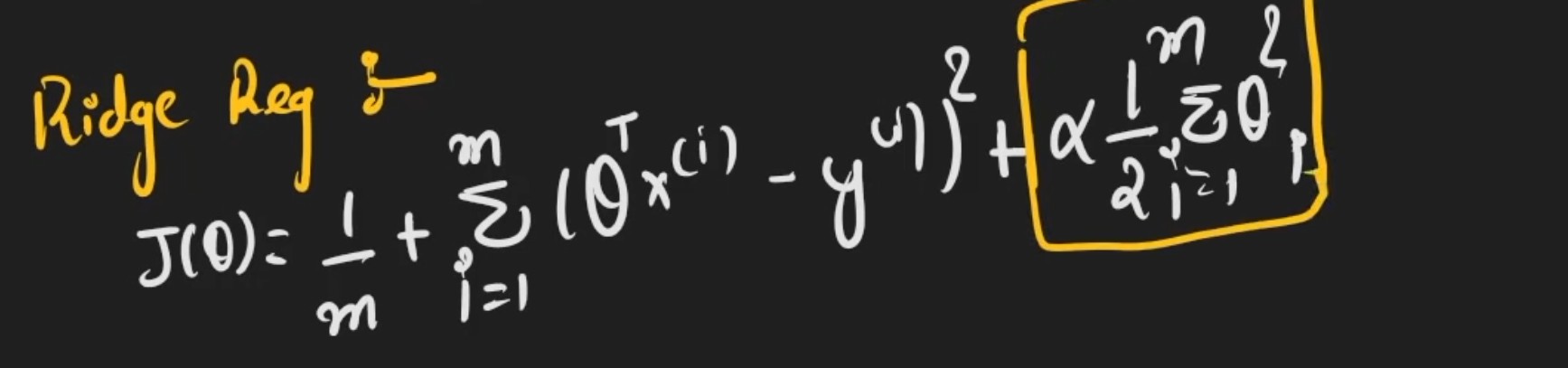
Eliminating theta

//

Regularization will help one to eliminate the features which are less helpful

Ridge Regression: Add a regularization term to the end of the cost function.

Ex.



Regularization Term / L2 norm

What the pointed to eq variable in the example above dose is lower the weight of unwanted features and brings them closer to 0. Aka the theta is penalized and assigned a lower weight in the model created. The **α variable (lambda) contains** how strict to be on the feature. This eq only goes to every column of data after 0 because 0 is our bias.

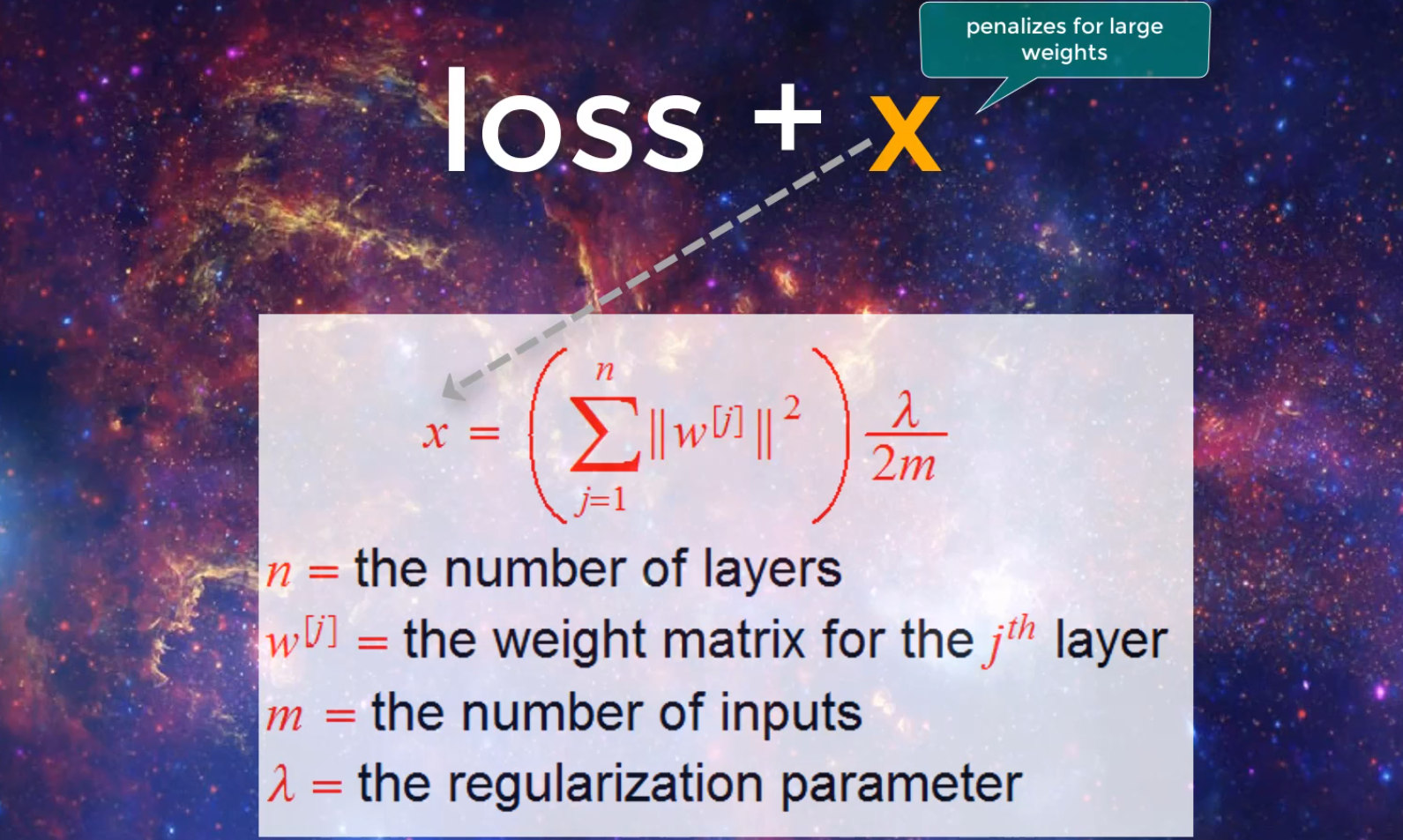
Method A: Regularization

Regularization will help will lowering the Overfitting problem when it comes to training datasets. Usually done by penalizing for complexity (theta). The data in our models may not be able to generalize well because of over trained data. The exchange that takes place in the model when using regularization models are fit training data for the increate of generalizing data that’s been unseen(predication).

↓

To use regularization in our data models we need to add a variable that penalized for large weights in our cost equations. Ex …. (loss + x)

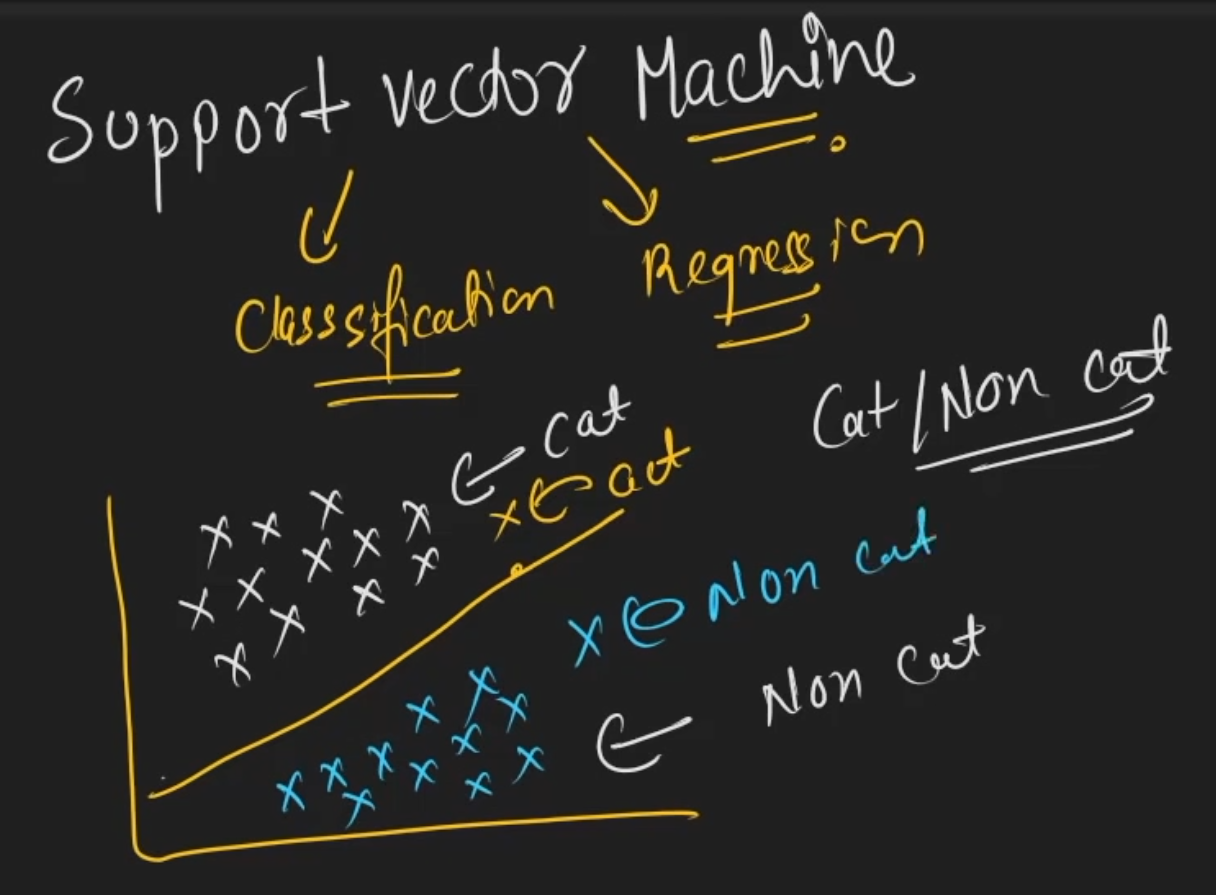
L2



The sum of the squared norms of the weight matrix multiplied by a small constant. Norms is a function that assigns a positive link or size to a vector or vector space. The vector space depends on the weight of our matrixes (going to be a positive / > 0). As the filters begins to weight grossly over 0 we can take these now crappy filters and get rid of them making out models more efficient and correct.

Support Vector Machine

Is a supervised learning algo, which is a boat for **classification** (Logistic Regression) and (linear regression) **Regression**.



* The SVM model takes two classified dataset and construct a two defend hyperplanes and will also construct a margin between the hyperplanes and dataset. The goal is to construct this hyperplane in such a way that the nearest datapoint(x^i) that most related to the data set is far away from the hyperplane.



* The support vectors that are going to lie between the hyperplane and corresponding dataset are going to be your classifying body so if a vector(datapoint/xi) is past this supporting vector then it’ll be classified accordingly.

Hard Margin: NO data points to come into that margin (consequently will cause overfitting)

Soft Margin: