Machine learning


why do you think I'm learning AI?
to get a job
no, to understand memes

What is Machine learning ?

Machine Learning is the science (and art) of programming computers so they can learn from data.

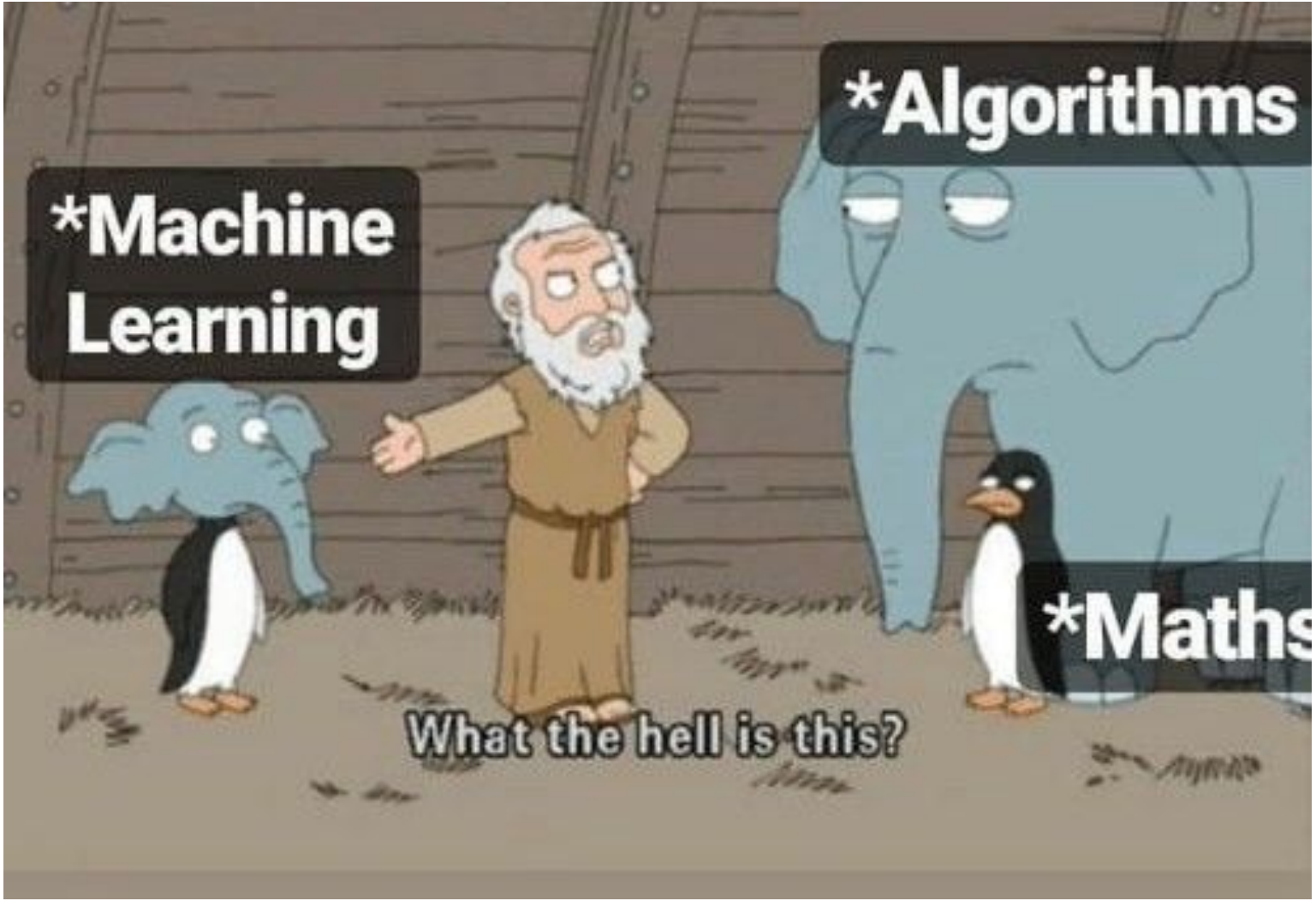
Here is a slightly more general definition:

[Machine Learning is the] field of study that gives computers the ability to learn without being explicitly Programmed. Arthur Samuel, 1959

And a more engineering-oriented one:

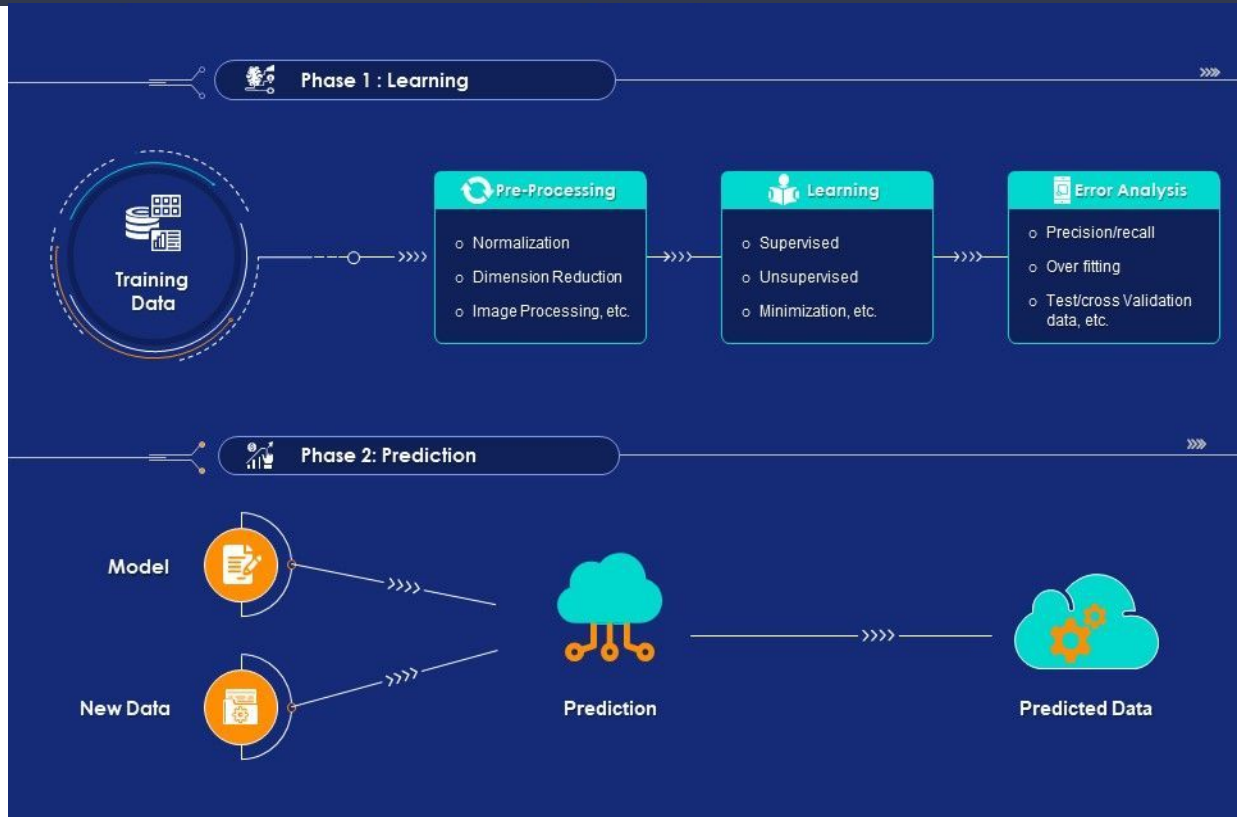
A computer program is said to learn from experience E with respect to some task T and some performance measure P, if its performance on T, as measured by P, improves with experience E.

Tom Mitchell, 1997


*Machine Learning
*Algorithms
*Maths
What the hell is this?

What is Machine learning ?

Machine Learning provides smart alternatives to analyzing vast volumes of data. By developing fast and efficient algorithms and data-driven models for real-time processing of data, Machine Learning can produce accurate results and analysis.



Type Of Machine Learning Problems?

Supervised

Unsupervised

Reinforcement

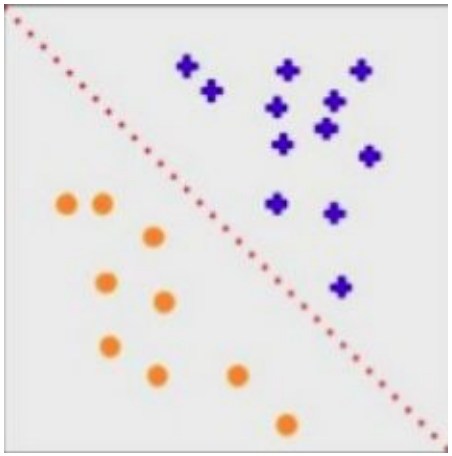
Supervised

Learn through examples of which we know the desired output(what we want to predict).

- **Is this a cat or a dog ?**
- **Are these emails spam or not?**
- **Predict the market value of houses given the square meters,number of rooms , neighborhood ,etc.**

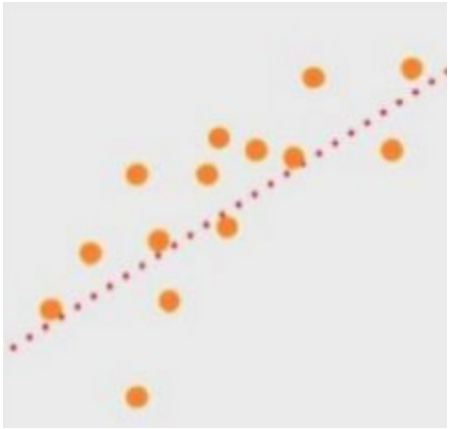
Classification

Output is a **discrete** Variable (e.g.,cat/dog)



Regression

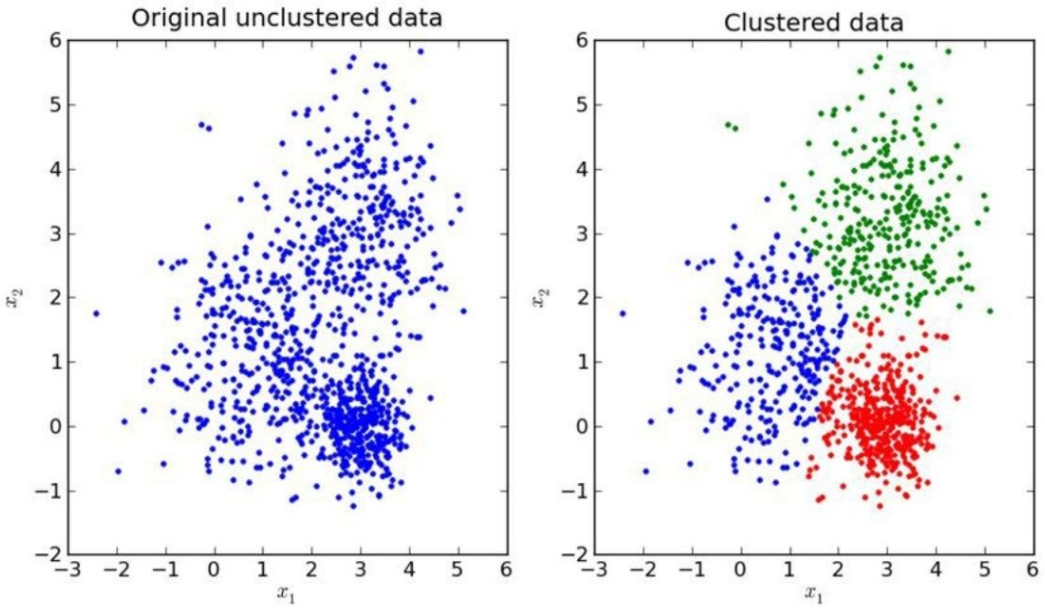
Output is **continuous** (e.g.,price ,temperature)



Unsupervised

There is no desired output .Learn something about the data.Latent relationships.

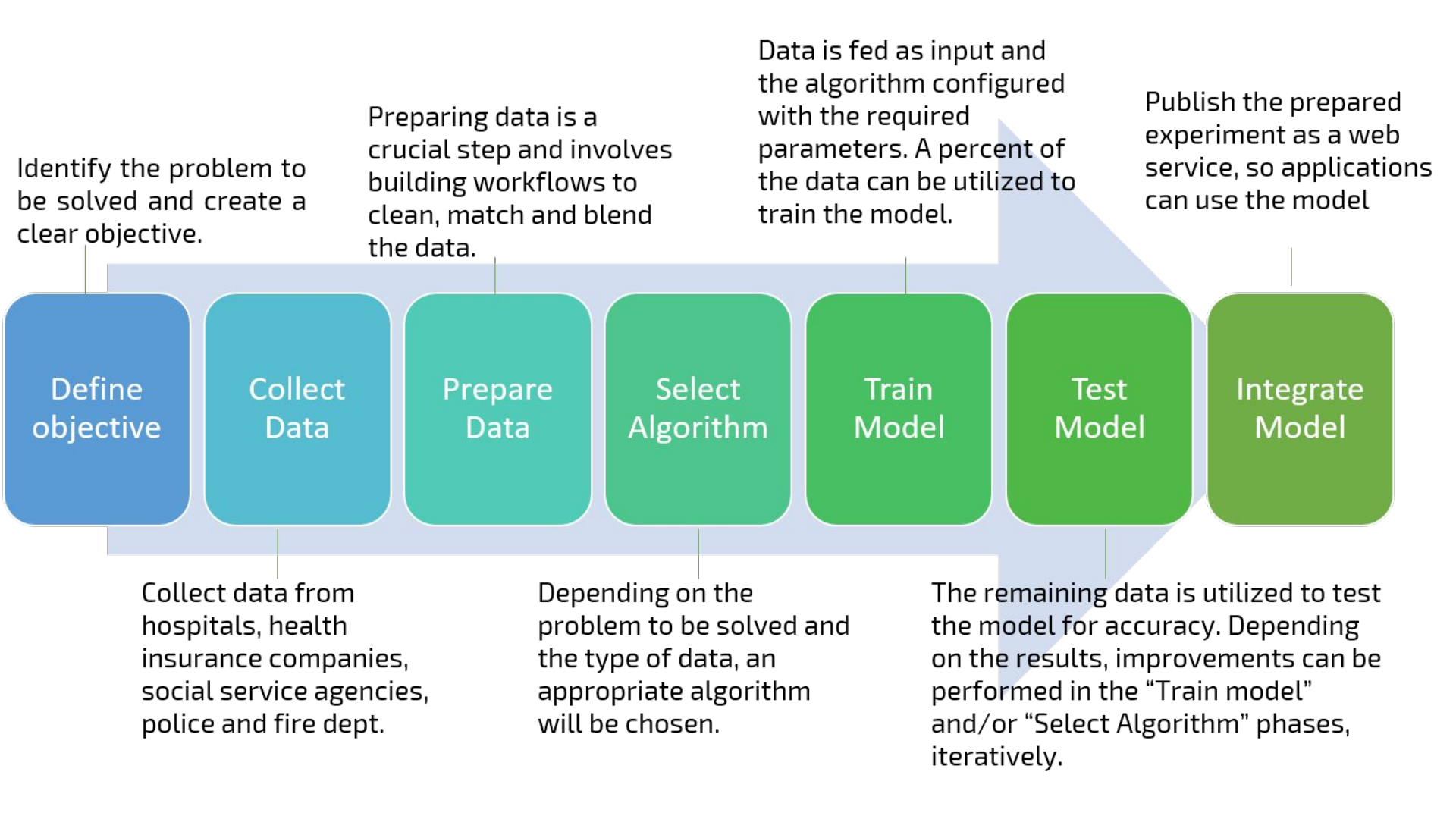
- **I have photos and want to put them in 20 groups**
- **I want to find anomalies in the credit card usage of patterns of my customers.**



Reinforcement

An agent **interacts** with an **environment** and watches the result of the interaction

Environment gives feedback via positive or negative **reward signal**.


Identify the problem to be solved and create a clear objective.
Preparing data is a crucial step and involves building workflows to clean, match and blend the data.
Data is fed as input and the algorithm configured with the required parameters. A percent of the data can be utilized to train the model.
Publish the prepared experiment as a web service, so applications can use the model
Define objective
Collect Data
Prepare Data
Select Algorithm
Train Model
Test Model
Integrate Model
Collect data from hospitals, health insurance companies, social service agencies, police and fire dept.
Depending on the problem to be solved and the type of data, an appropriate algorithm will be chosen.
The remaining data is utilized to test the model for accuracy. Depending on the results, improvements can be performed in the “Train model” and/or “Select Algorithm” phases, iteratively.

Data Gathering

Might depend on human work

- **Manual labeling for supervised learning**
- **Domain knowledge.Maybe even experts**

May come for free or ‘ort of

- **Eg.,Machine translation**

The more the better .some algorithms need larger amounts of data to be useful (eg.,neural networks).

The quantity and quality of data dictate model accuracy

Data Preprocessing

Is there anything wrong with the data?

- **Missing values**
- **Outliers**
- **Bad encoding(for text**
- **Wrongly-labeled examples**
- **Biased data**

Need to fix /remove data?

Algorithm Selection & Training

Supervised

- Linear classifier
- Naive Bayes
- Support Vector Machines (SVM)
- Decision Tree
- Random Forests
- k-Nearest Neighbors
- **Neural Networks (Deep learning)**

Unsupervised

- PCA
- t-SNE
- k-means
- DBSCAN

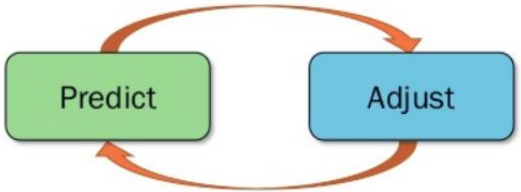
Reinforcement

- SARSA–λ
- Q-Learning

Algorithm Selection & Training

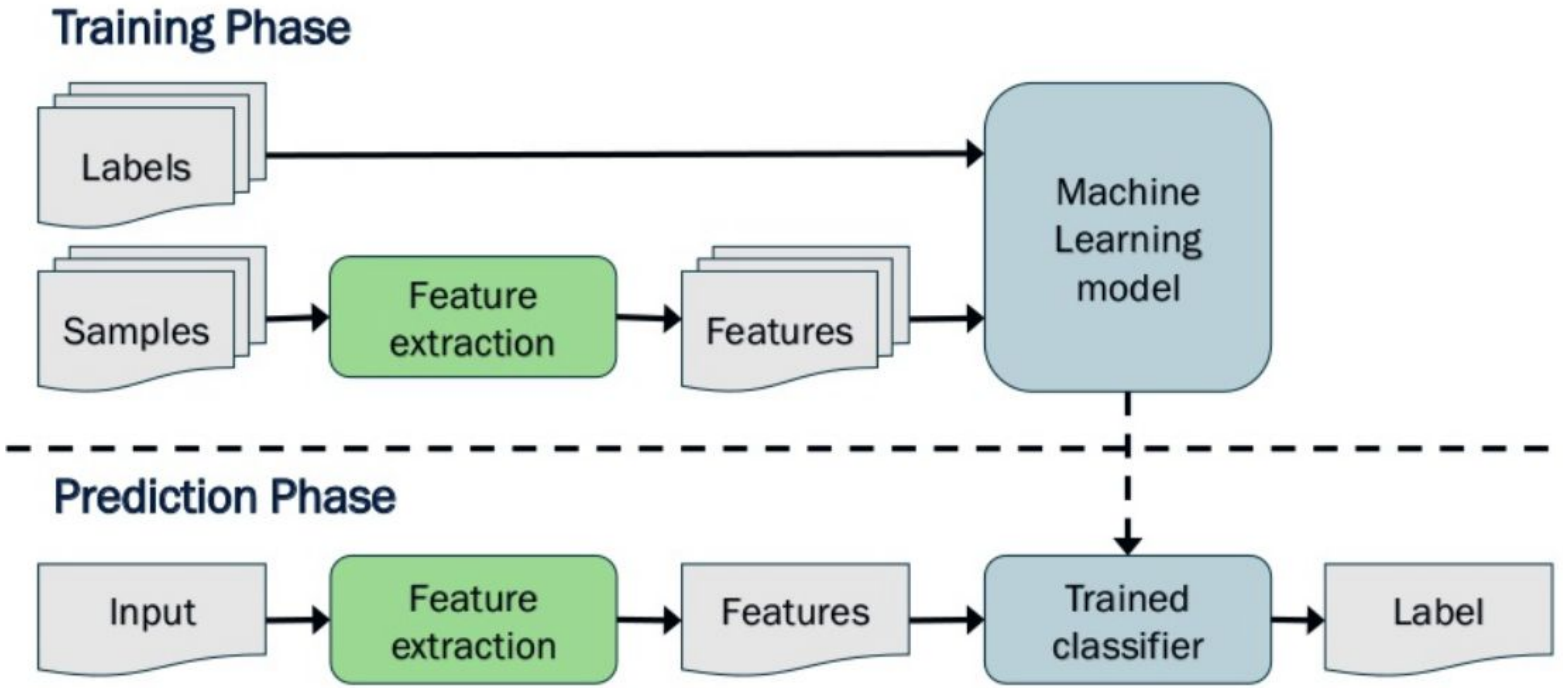
Goal of training : making the correct prediction as often possible

- **Incremental improvement :**



- **Use of metrics for evaluation performance and comparing solutions**
- **Hyperparameter tuning : more an art then a science**

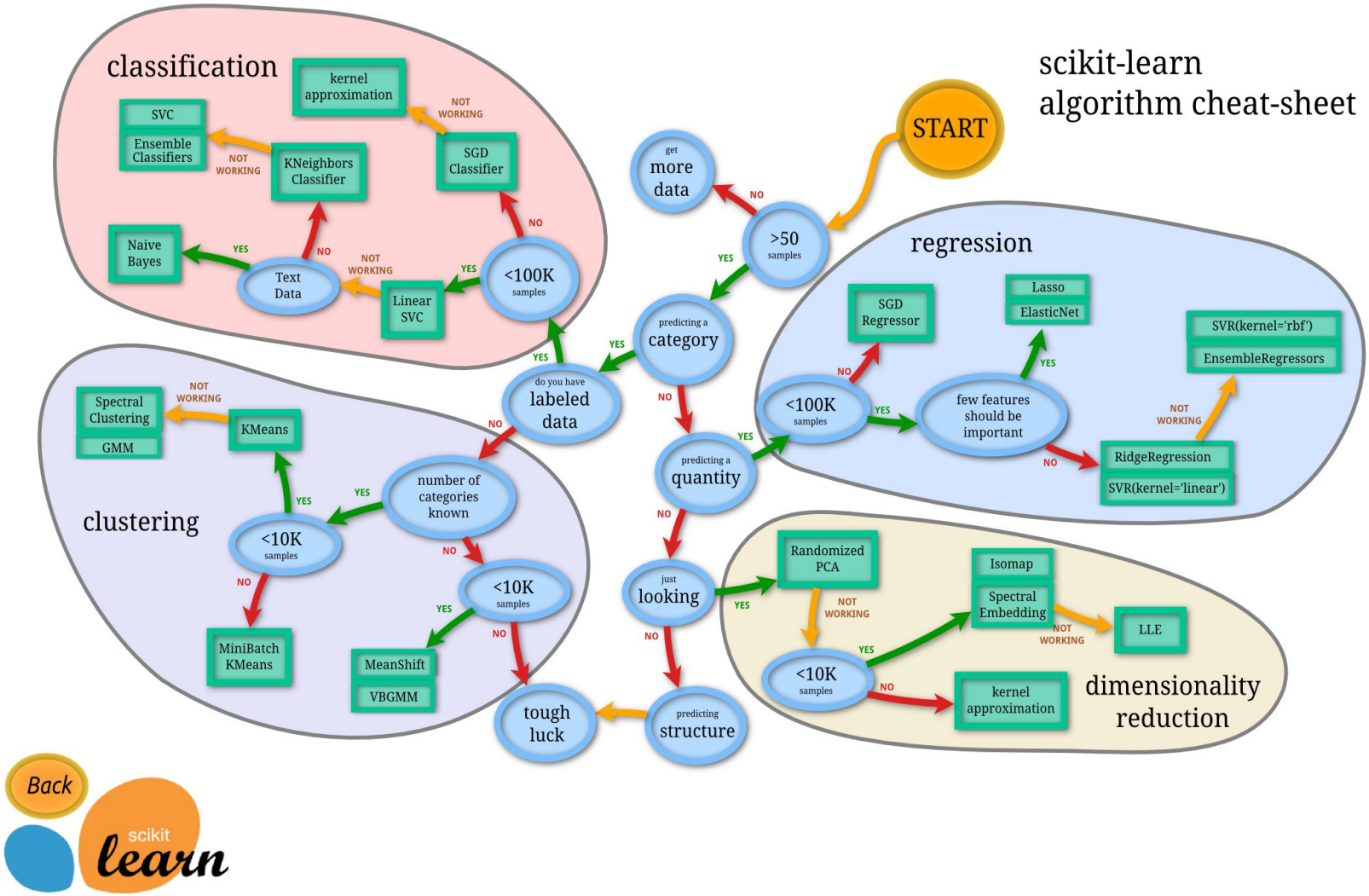
Making Predictions



Sklearn

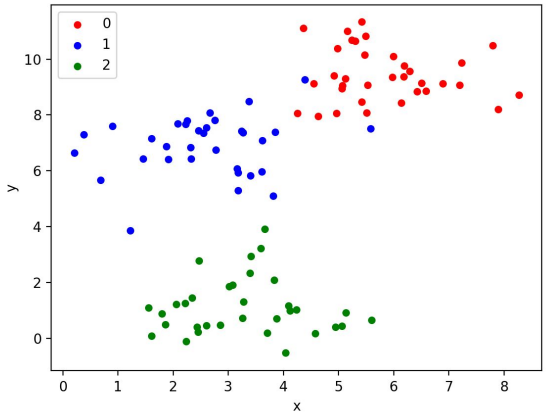
Sklearn

- **Simple and efficient tools for predictive data analysis**
- **Accessible to everybody, and reusable in various contexts**
- **Built on NumPy, SciPy, and matplotlib**
- **Open source, commercially usable - BSD license**


scikit-learn
algorithm cheat-sheet
START
get more data
>50 samples
predicting a category
do you have labeled data
predicting a quantity
just looking
predicting structure
tough luck
classification
kernel approximation
SVC
Ensemble Classifiers
KNeighbors Classifier
SGD Classifier
Naive Bayes
Text Data
Linear SVC
<100K samples
NOT WORKING
YES
NO
regression
SGD Regressor
Lasso
ElasticNet
SVR(kernel='rbf')
EnsembleRegressors
<100K samples
few features should be important
RidgeRegression
SVR(kernel='linear')
clustering
Spectral Clustering
GMM
KMeans
number of categories known
<10K samples
MiniBatch KMeans
MeanShift
VBGMM
dimensionality reduction
Randomized PCA
Isomap
Spectral Embedding
LLE
<10K samples
kernel approximation
Back
scikit learn

Datasets

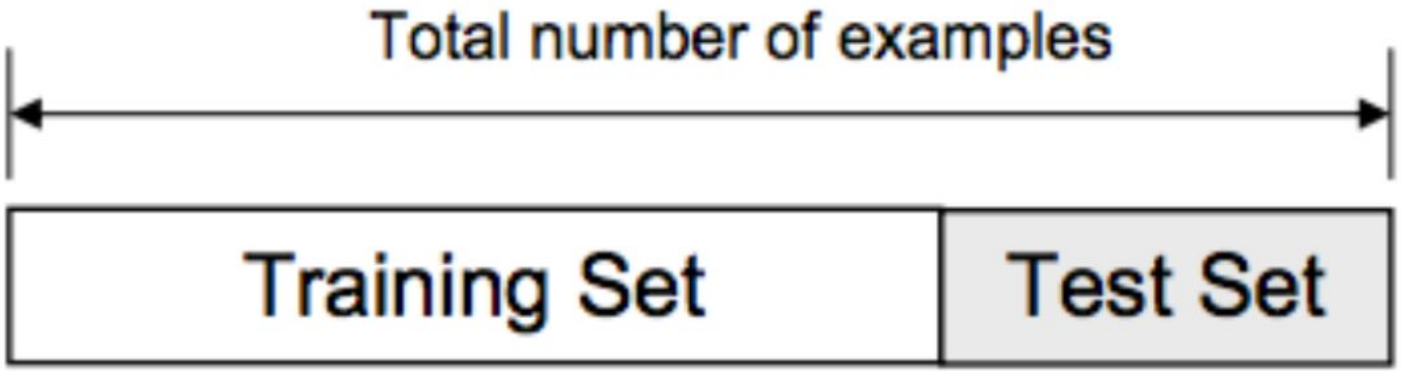
The sklearn.datasets module includes utilities to load datasets, including methods to load and fetch popular reference datasets. It also features some artificial data generators.




```
sklearn.datasets.load_boston()
```

Train test split

Split arrays or matrices into random train and test subsets




```
sklearn.model_selection.train_test_split(x,y)
```

Min-Max normalization:

It is simple way of scaling values in a column. But, it tries to move the values towards the mean of the column

$$z = \frac{x \text{ - } \min(x)}{\max(x) \text{ - } \min(x)}$$

```
sklearn.preprocessing.MinMaxScaler()
```

StandardScaler

Standardize features by removing the mean and scaling to unit variance

$$z = \frac{x - \mu}{\sigma}$$

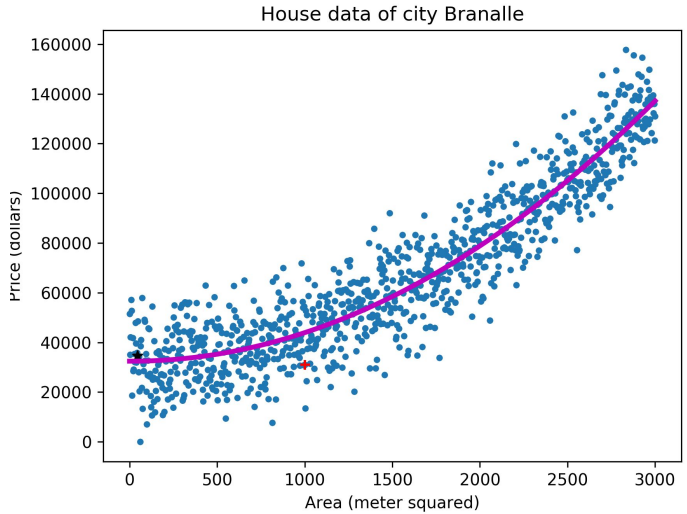
μ = Mean

σ = Standard Deviation

```
sklearn.preprocessing.StandardScaler(*, copy=True, with_mean=True, with_std=True)
```

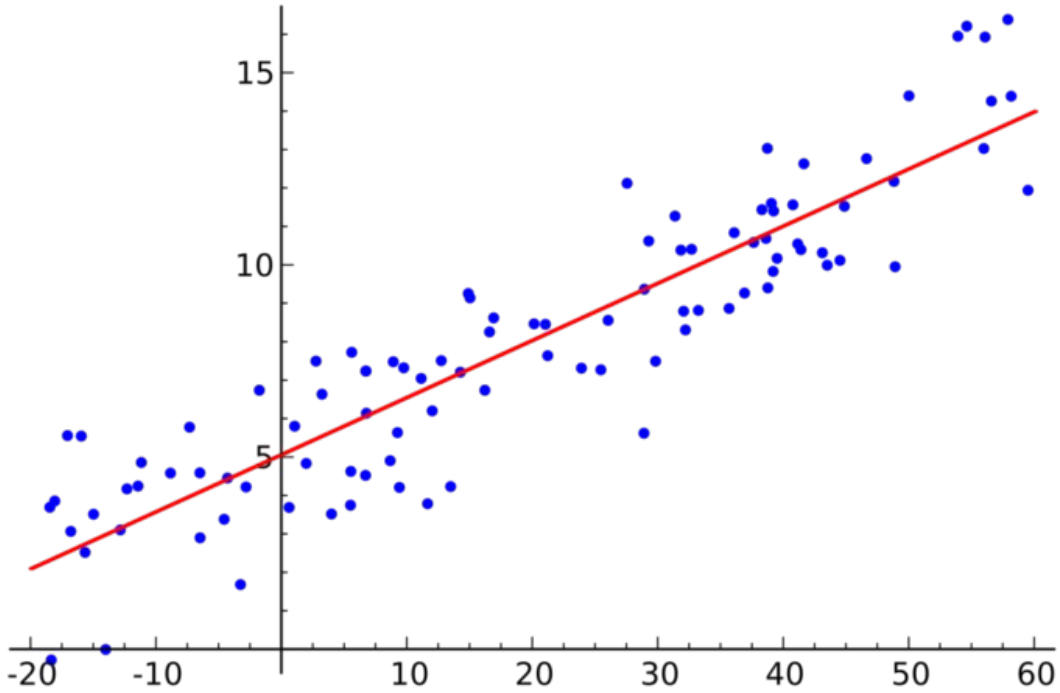
Regression

Regression analysis consists of a set of machine learning methods that allow us to predict a continuous outcome variable (y) based on the value of one or multiple predictor variables (x). Briefly, the goal of regression model is to build a mathematical equation that defines y as a function of the x variables.



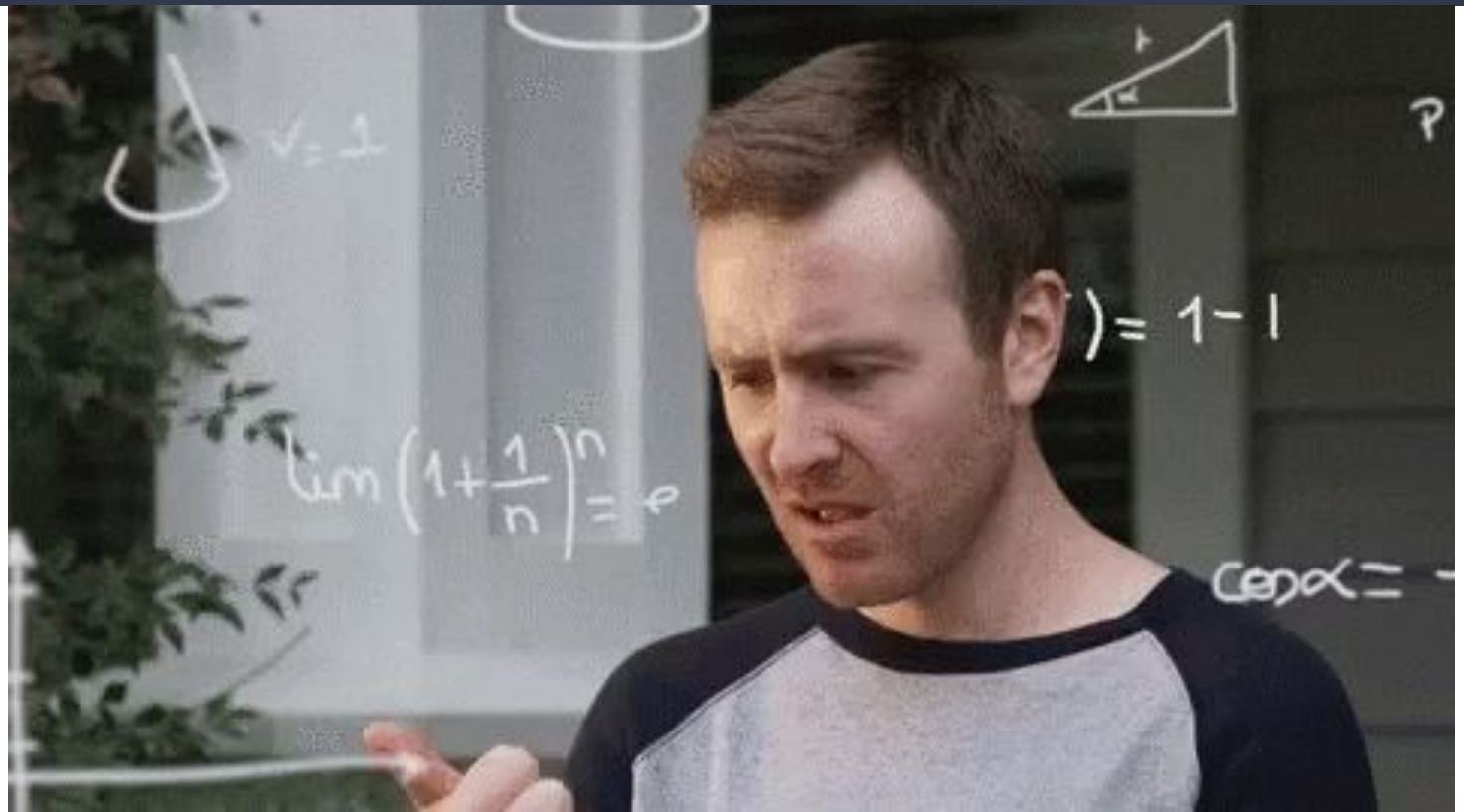
Linear Regression

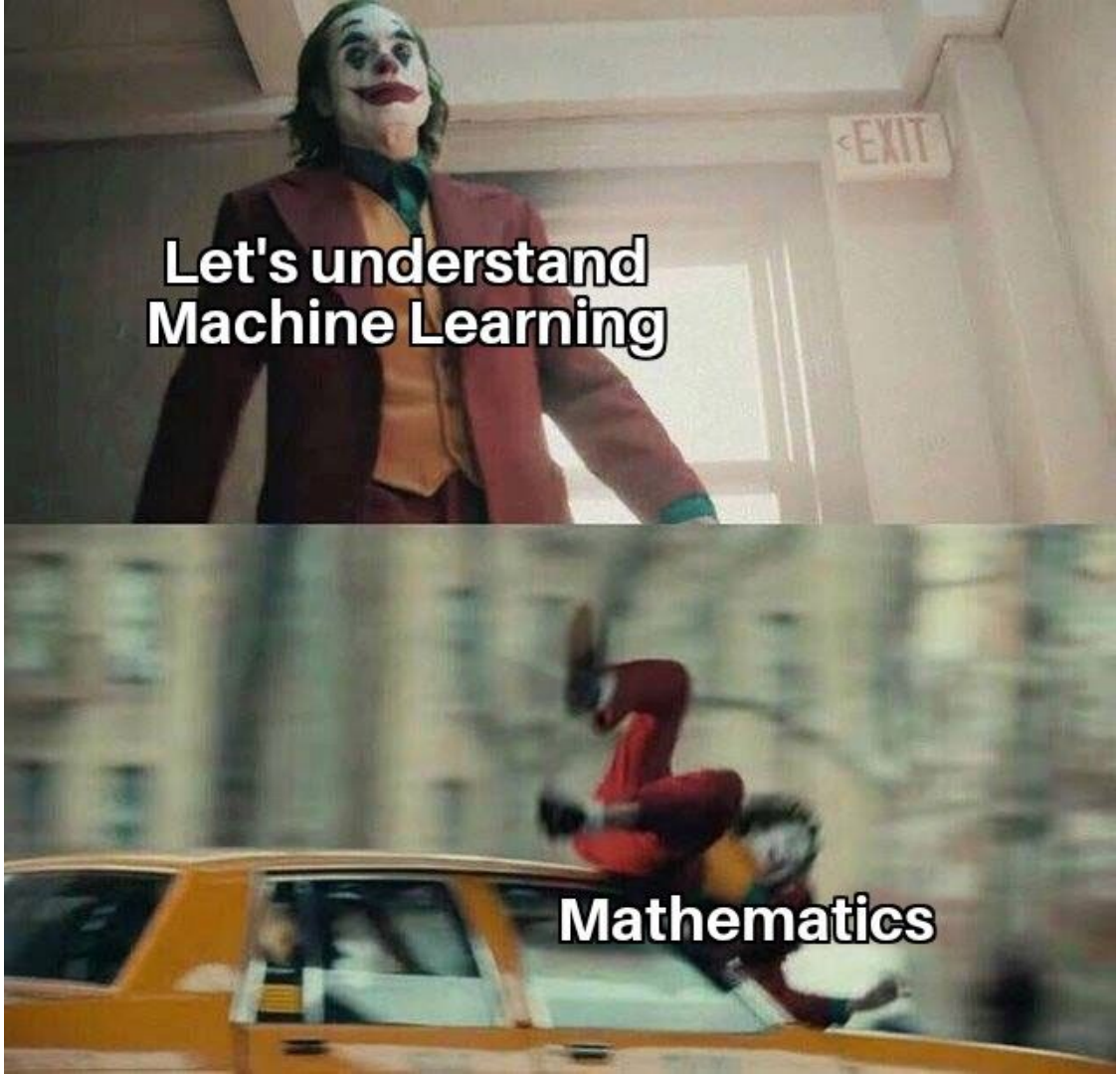
Simple linear regression is a type of regression analysis where the number of independent variables is one and there is a linear relationship between the independent(x) and dependent(y) variable.



$$y = b_0 + W_1 * X_1$$

Linear Regression




EXIT
Let's understand
Machine Learning
Mathematics

Linear Regression

The motive of the linear regression algorithm is to find the best values for W_n and b_n

Simple Linear Regression

$$y = b_0 + W_1 * X_1$$

Multiple Linear Regression

$$y = b_0 + W_1 * X_1 + W_2 * X_2 + W_n * X_n$$

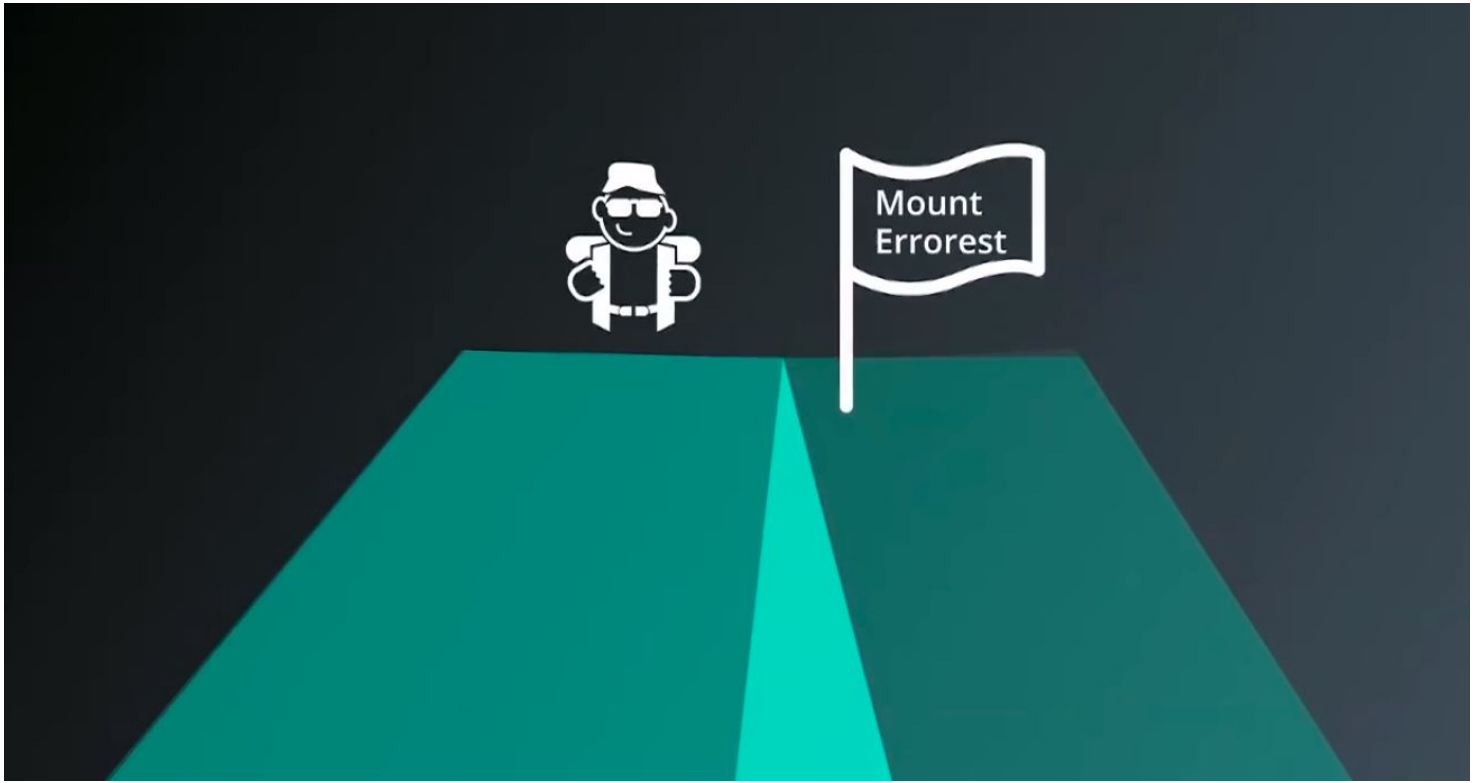

Dog

Cost Function

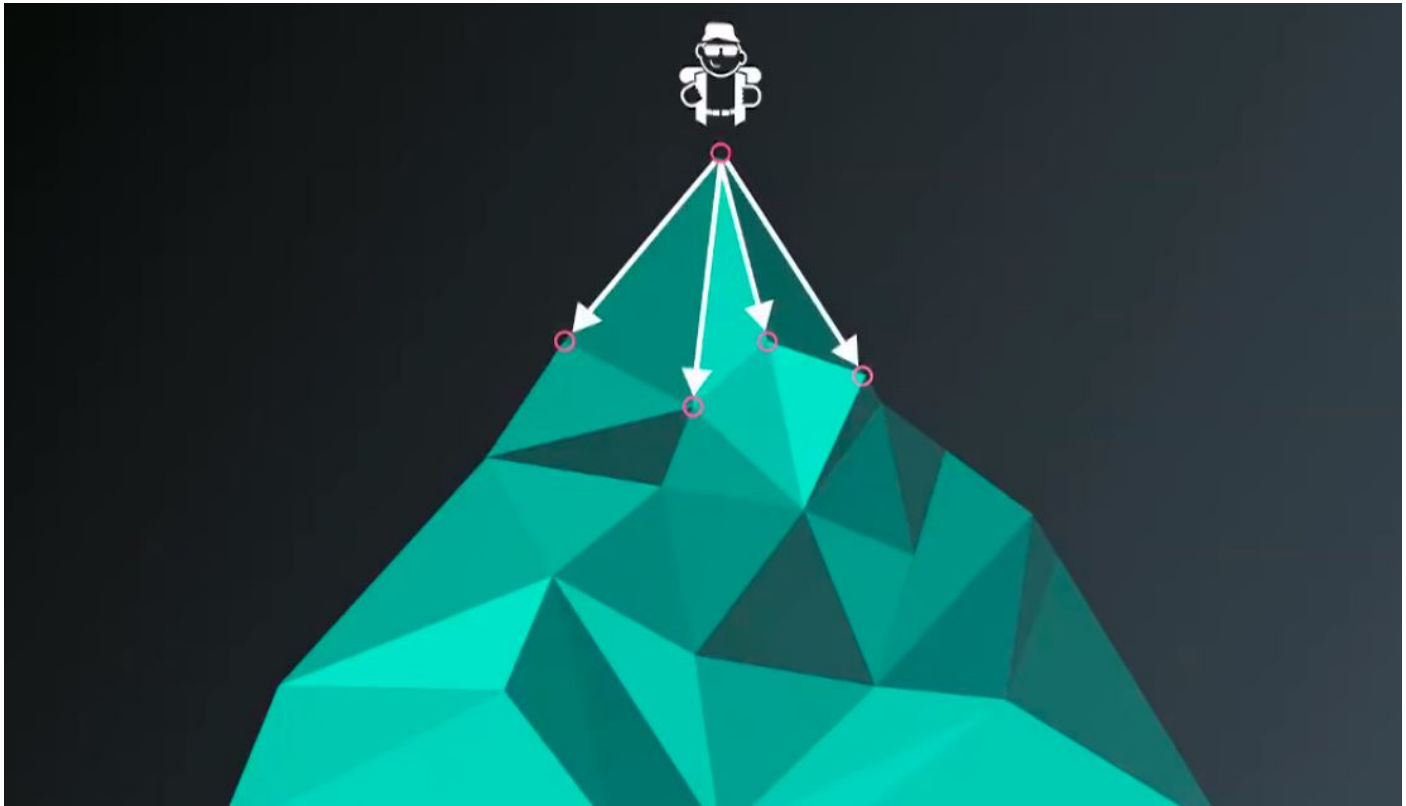
The cost function helps us to figure out the best possible values for θ_0 and θ_1 which would provide the best fit line for the data points. Since we want the best values for θ_0 and θ_1, we convert this search problem into a minimization problem where we would like to minimize the error between the predicted value and the actual value.

$$J(\theta_0, \theta_1) = \frac{1}{2m} \sum_{i=1}^{m} \left(h_\theta\left(x^{(i)}\right) - y^{(i)} \right)^2$$

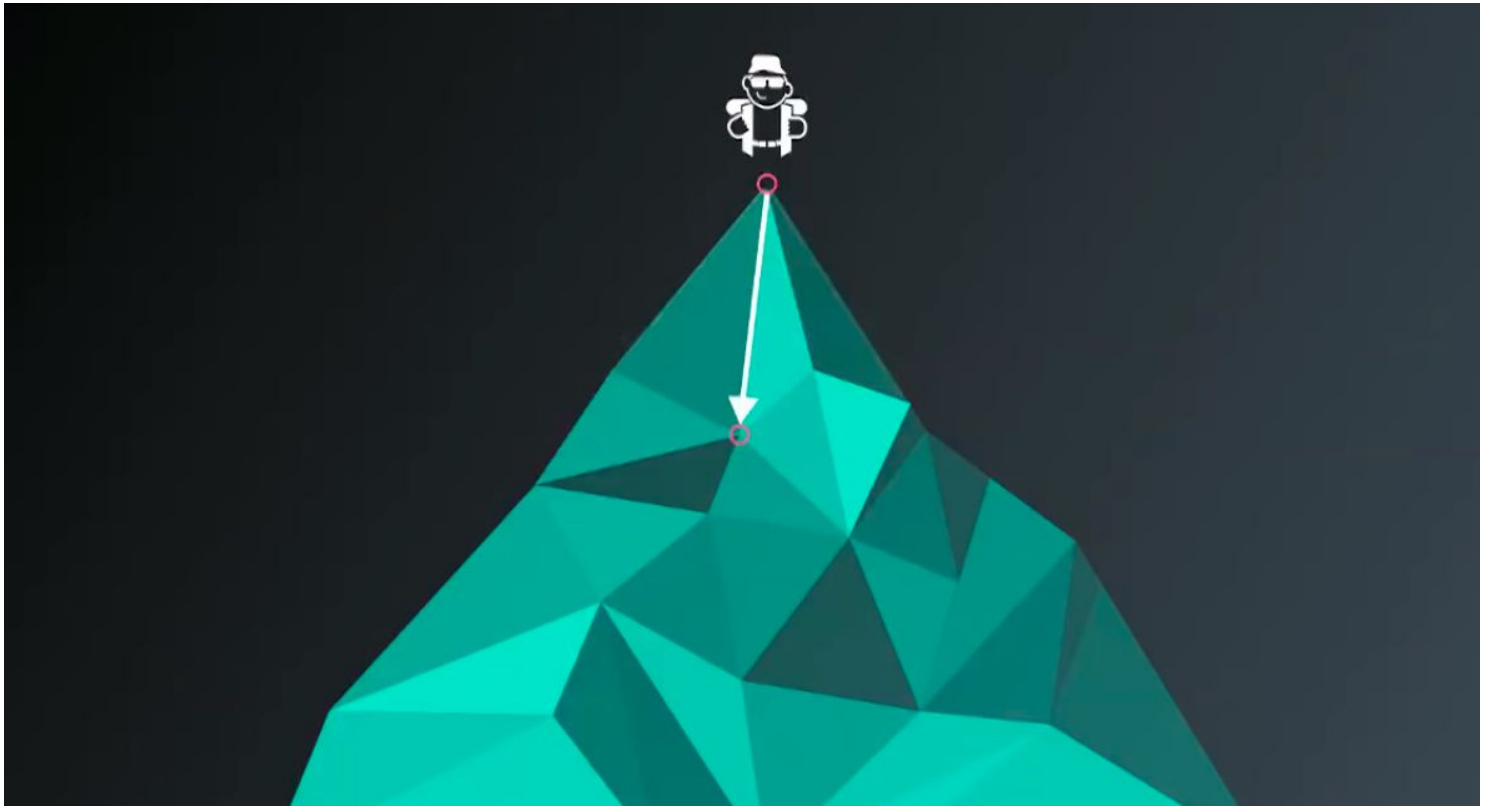
Loss Optimization



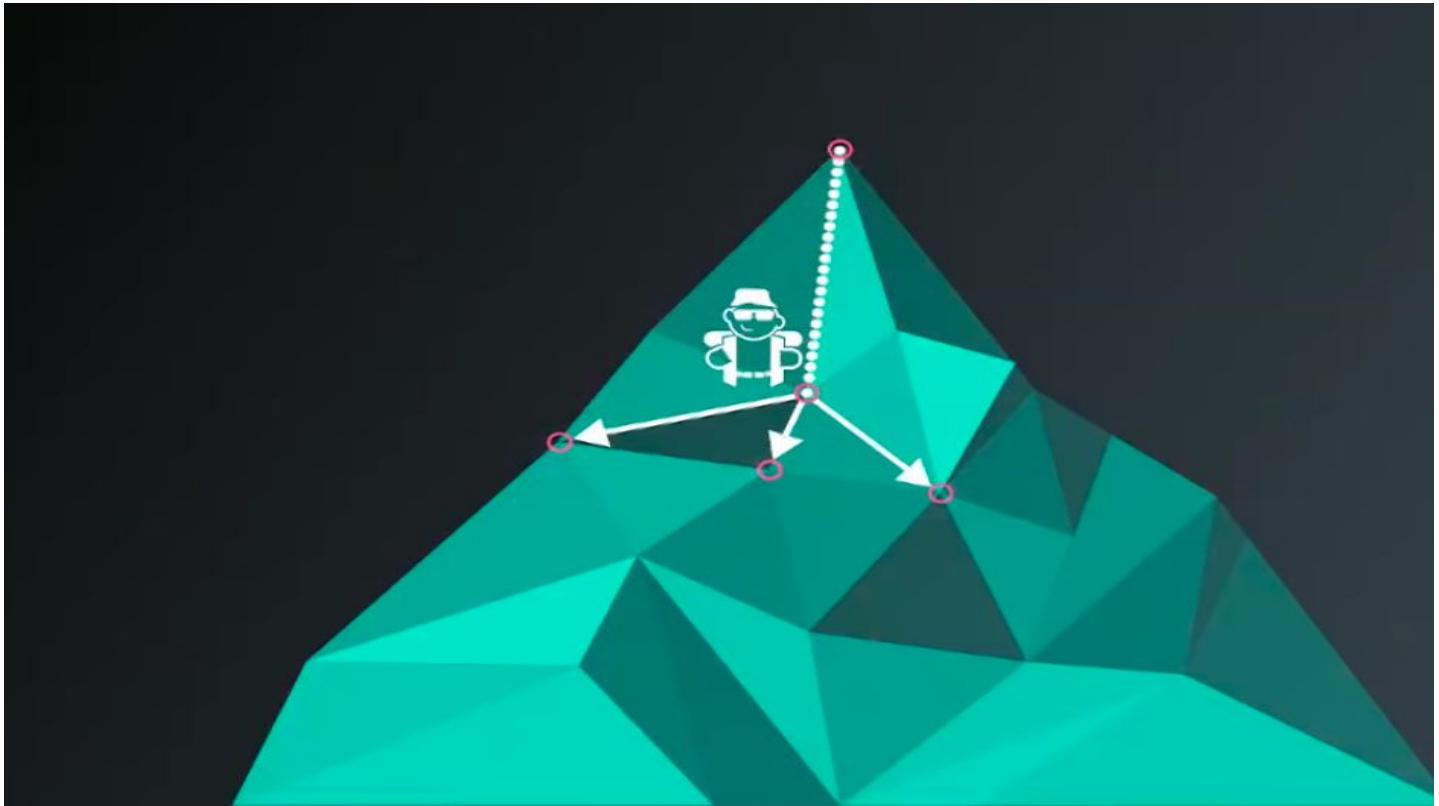
Loss Optimization



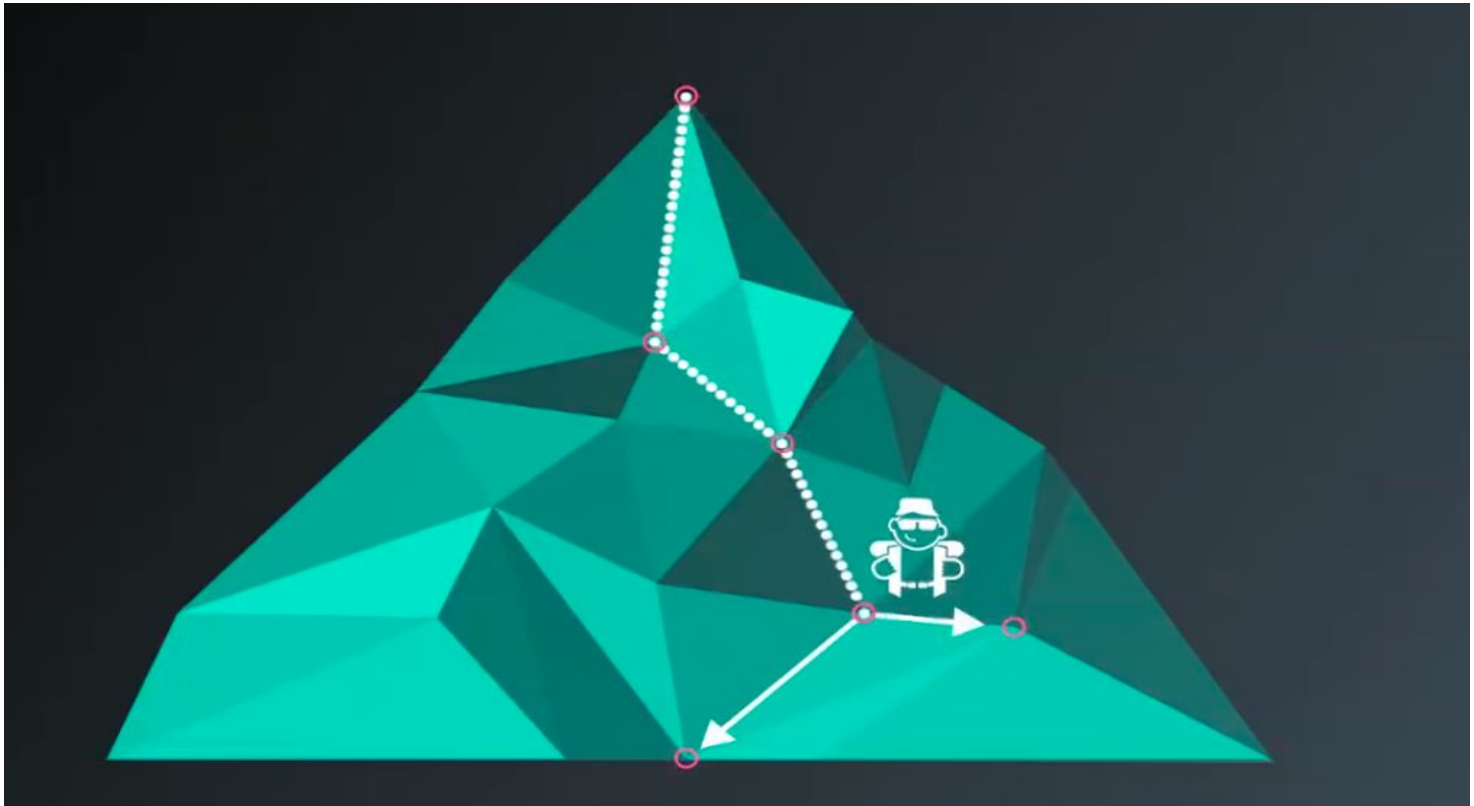
Loss Optimization



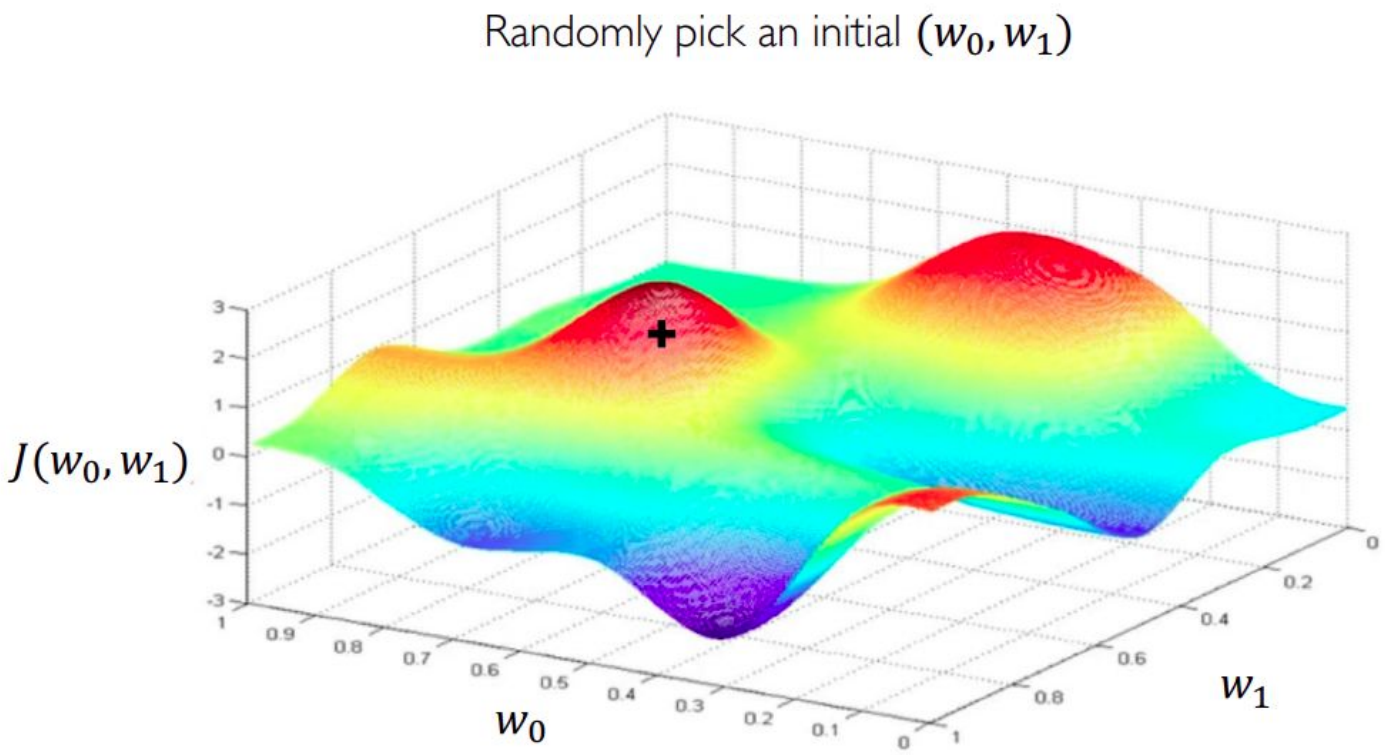
Loss Optimization



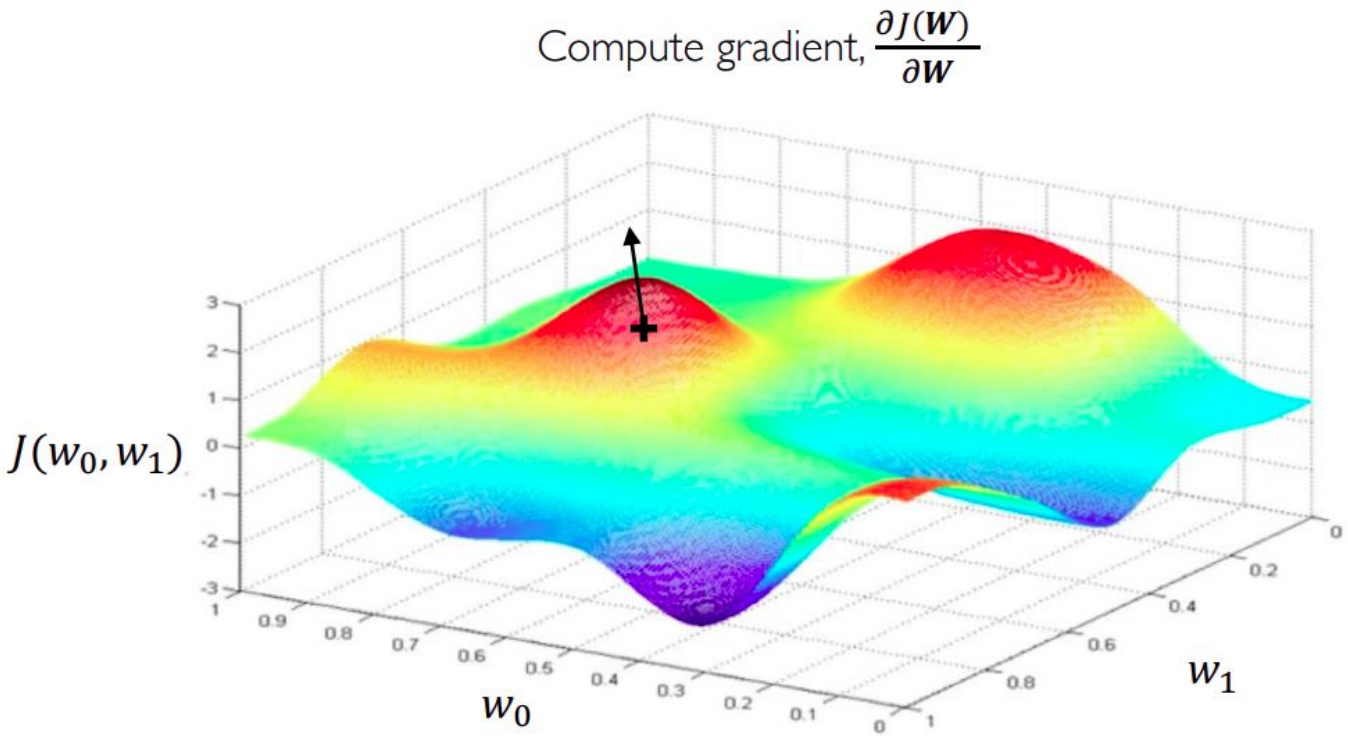
Loss Optimization



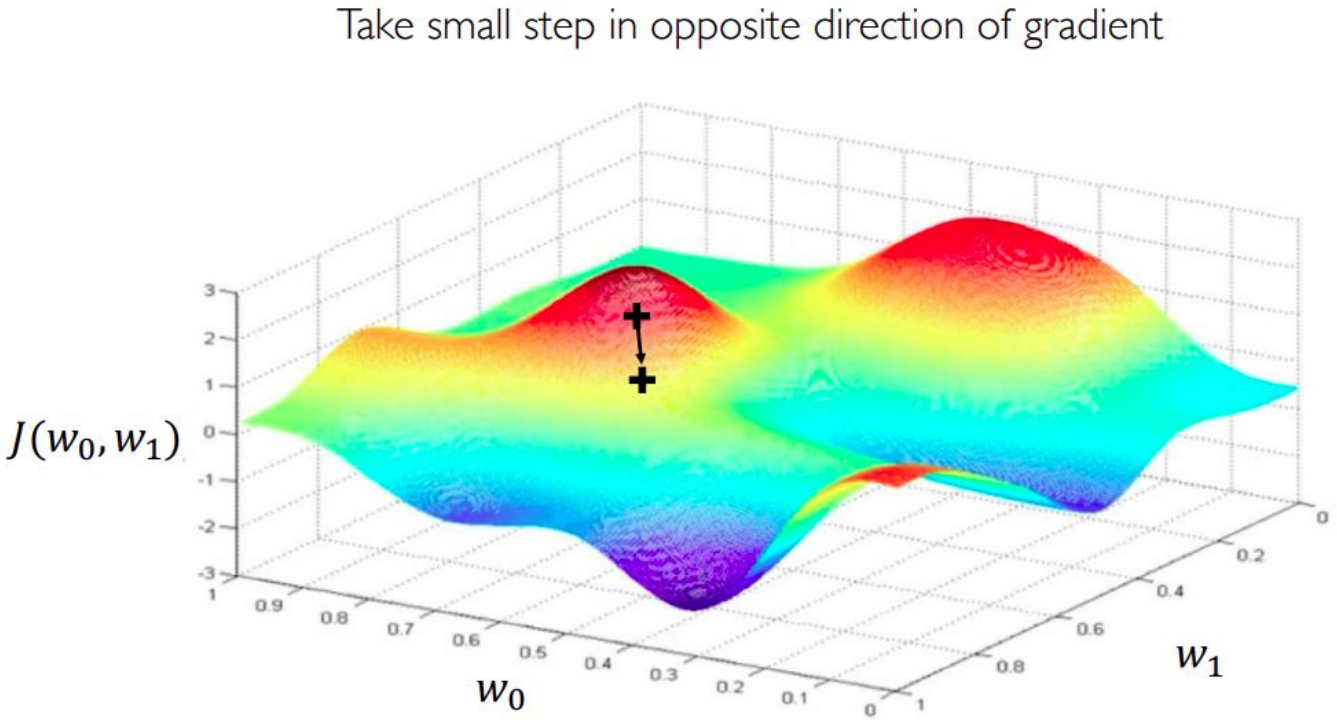
Loss Optimization



Loss Optimization

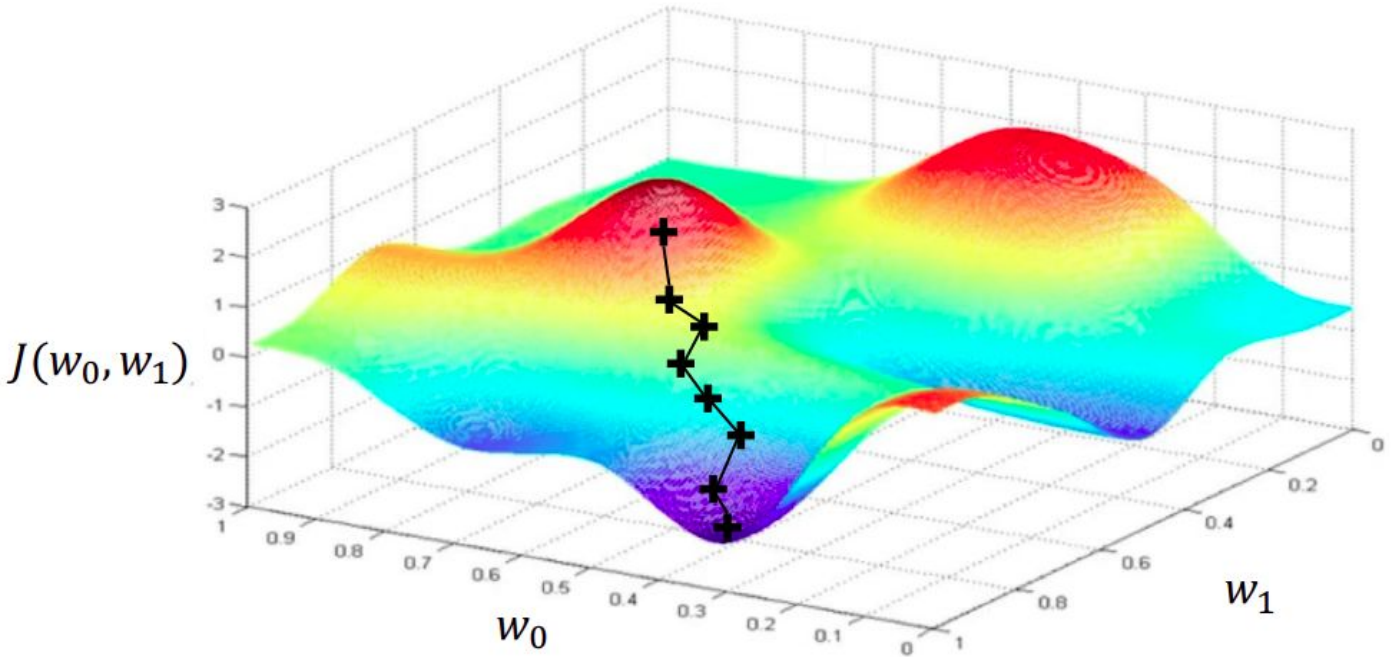


Loss Optimization



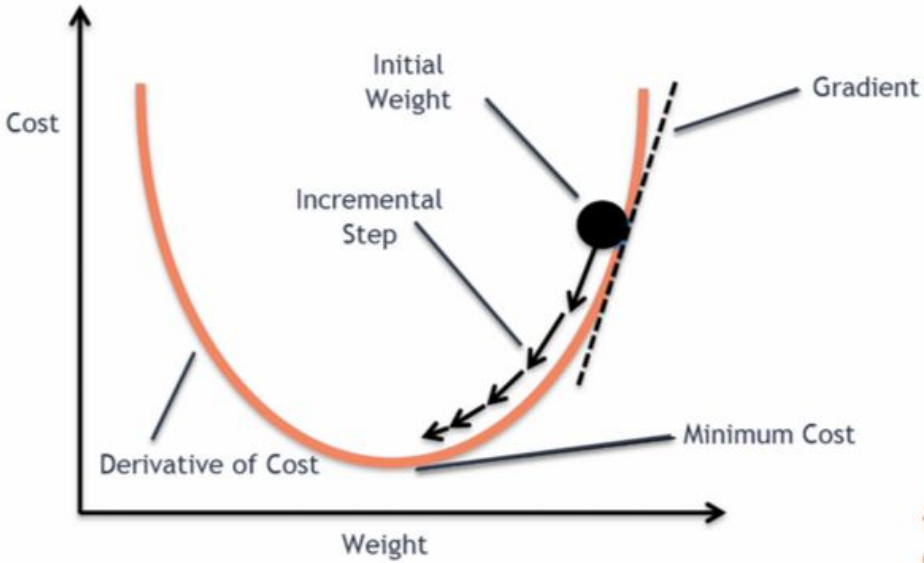
Gradient Descent

Repeat until convergence



Gradient Descent

Gradient descent is a method of updating b_n to reduce the cost function(MSE),The idea is that we start with some values for b_n and then we change these values iteratively to reduce the cost. Gradient descent helps us on how to change the values.



Gradient Descent

Gradient descent is a method of updating bn to reduce the cost function(MSE),The idea is that we start with some values for bn and then we change these values iteratively to reduce the cost. Gradient descent helps us on how to change the values.

Hypothesis:

$$h_\theta(x) = \theta_0 + \theta_1 x$$

Parameters:

$$\theta_0, \theta_1$$

Cost Function:

$$J(\theta_0, \theta_1) = \frac{1}{2m} \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)})^2$$

Goal:

$$\underset{\theta_0, \theta_1}{minimize} J(\theta_0, \theta_1)$$

Gradient Descent

Gradient descent is a method of updating bn to reduce the cost function(MSE),The idea is that we start with some values for bn and then we change these values iteratively to reduce the cost. Gradient descent helps us on how to change the values.

Cost Function – "One Half Mean Squared Error":

$$J(\theta_0, \theta_1) = \frac{1}{2m}\sum_{i=1}^{m}\left(h_\theta\left(x^{(i)}\right) - y^{(i)}\right)^2$$

Objective:

$$\min_{\theta_0, \theta_1} J(\theta_0, \theta_1)$$

Update rules:

$$\theta_0 := \theta_0 - \alpha \frac{d}{d\theta_0} J(\theta_0, \theta_1)$$

$$\theta_1 := \theta_1 - \alpha \frac{d}{d\theta_1} J(\theta_0, \theta_1)$$

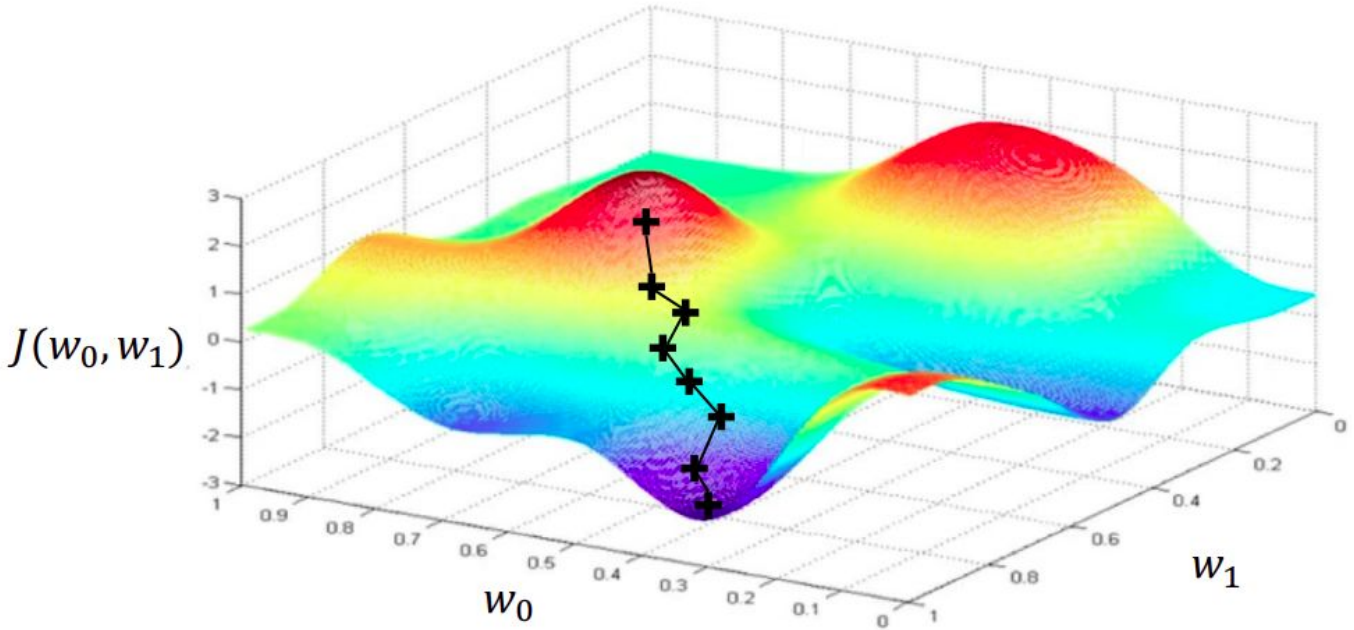
Derivatives:

$$\frac{d}{d\theta_0} J(\theta_0, \theta_1) = \frac{1}{m}\sum_{i=1}^{m}\left(h_\theta\left(x^{(i)}\right) - y^{(i)}\right)$$

$$\frac{d}{d\theta_1} J(\theta_0, \theta_1) = \frac{1}{m}\sum_{i=1}^{m}\left(h_\theta\left(x^{(i)}\right) - y^{(i)}\right) \cdot x^{(i)}$$

Gradient Descent

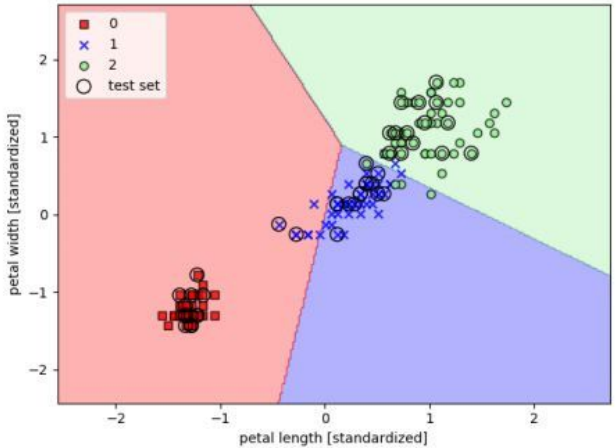
Repeat until convergence



Code

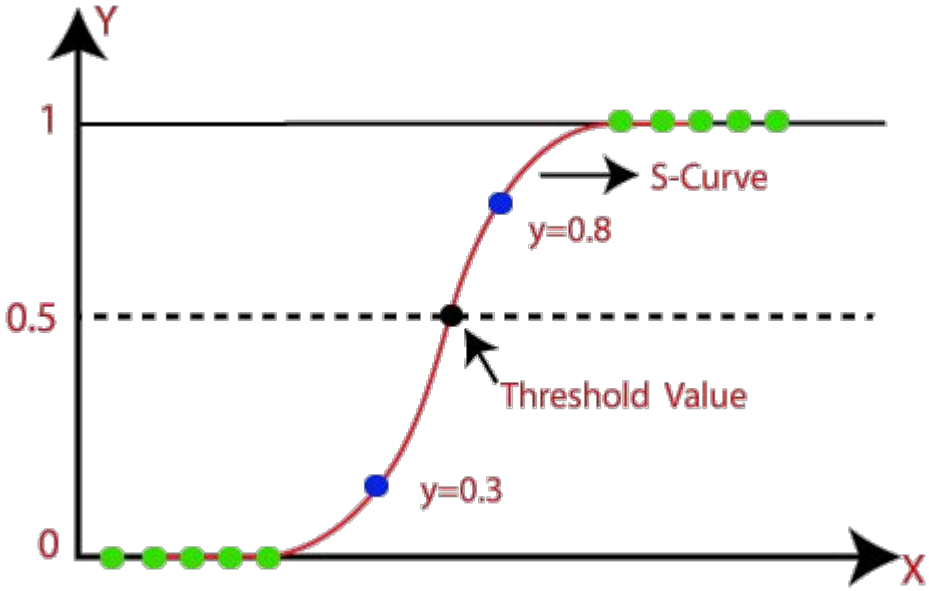
classification

classification refers to a predictive modeling problem where a class label is predicted for a given example of input data. Examples of classification problems include: Given an example, classify if it is spam or not.



Logistic Regression

Logistic regression is a classification algorithm used to assign observations to a discrete set of classes. Some of the examples of classification problems are Email spam or not spam, Online transactions Fraud or not Fraud, Tumor Malignant or Benign. Logistic regression transforms its output using the logistic sigmoid function to return a probability value.



Logistic Regression

Types of Logistic Regression

Binary Logistic Regression

Multinomial Logistic Regression

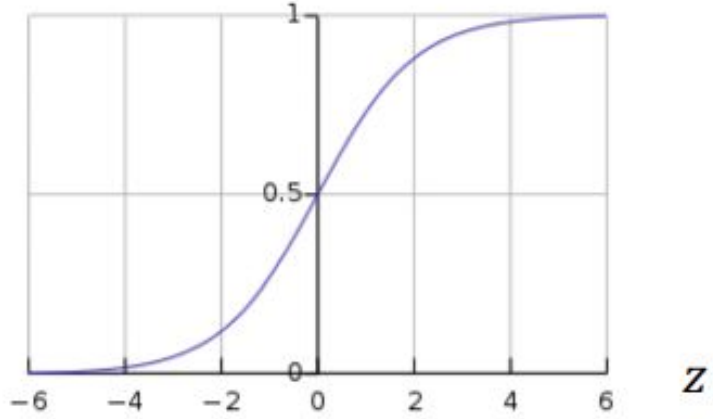
Binary Logistic Regression

The simplest form of logistic regression is binary or binomial logistic regression in which the target or dependent variable can have only 2 possible types either 1 or 0. I. In case of logistic regression, the linear function is basically used as an input to another function such as g in the following relation

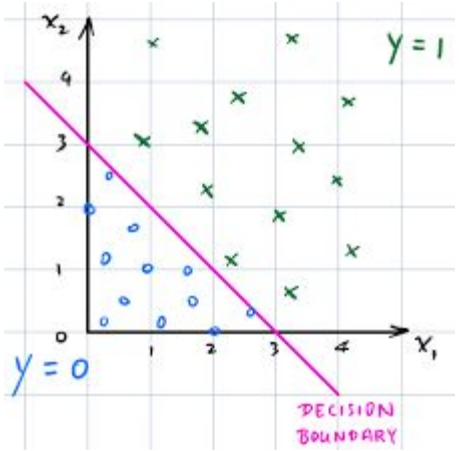
$$Z = g(b_0 + W_1 * X_1)$$

- Example: sigmoid function

$$g(z) = \sigma(z) = \frac{1}{1 + e^{-z}}$$



Decision Boundary

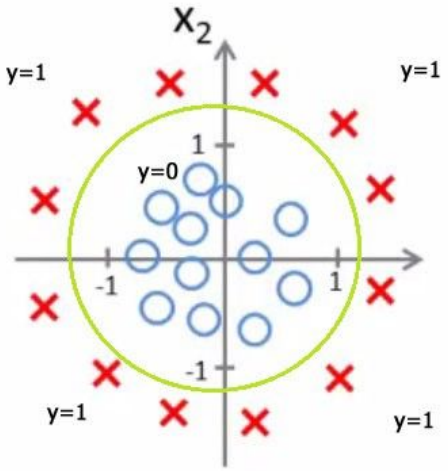


$$Z = g(b_0 + W_1 * X_1 + W_2 * X_2)$$

$$y = 1if - 3 + X_1 + X_2 \geq 0$$

$$X_1 + X_2 \geq 3$$

Decision Boundary



$$Z = g(b_0 + W_1 * X_1 + W_2 * X_2 + W_3 * X_1^2 + W_4 * X_2^2)$$

$$y = 1 if -1 + X_1^2 + X_2^2 \geq 0$$

$$X_1^2 + X_2^2 \geq 1$$

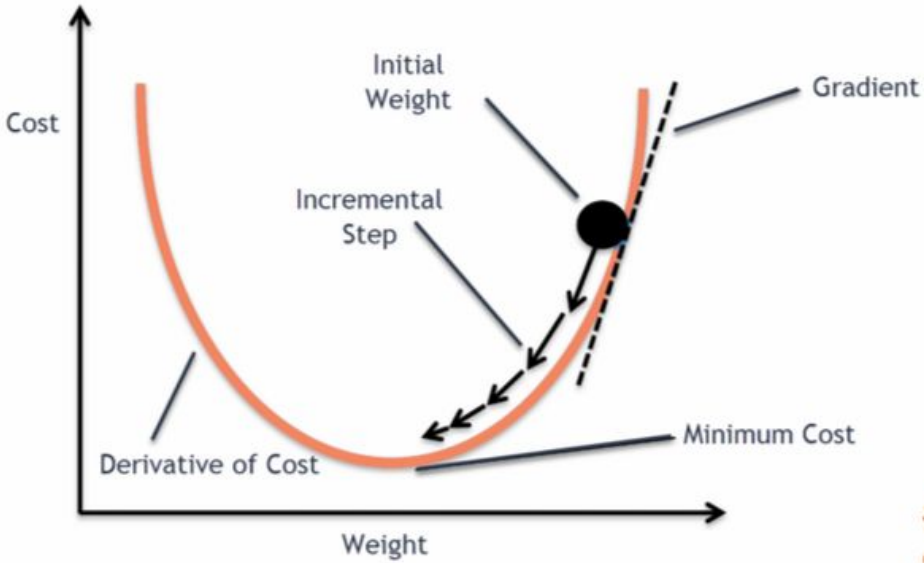

Dog

Cost Function

$$J(W) = -\frac{1}{m}(\sum_{i=1}^{m} y^i log(Z(x^i))) + (1 - y^i)(1 - log(Z(x^i))))$$

Gradient Descent

Gradient descent is a method of updating W_n to reduce the cost function,The idea is that we start with some values for b_n and then we change these values iteratively to reduce the cost. Gradient descent helps us on how to change the values.



Gradient Descent

Gradient descent is a method of updating W_n to reduce the cost function,The idea is that we start with some values for b_n and then we change these values iteratively to reduce the cost. Gradient descent helps us on how to change the values.

Hypothesis:

$$Z = g(b_0 + W_1 * X_1)$$

Parameters:

$$b_0 \ W_1$$

Cost Function:

$$J(W) = -\frac{1}{m}(\sum_{i=1}^{m} y^i log(Z(x^i))) + (1 - y^i)(1 - log(Z(x^i))))$$

Goal:

$$minimize J(W)$$

Gradient Descent

Gradient descent is a method of updating Wn to reduce the cost function,The idea is that we start with some values for bn and then we change these values iteratively to reduce the cost. Gradient descent helps us on how to change the values.

Cost Function

$$J(W) = -\frac{1}{m}(\sum_{i=1}^{m} y^i log(Z(x^i))) + (1 - y^i)(1 - log(Z(x^i))))$$

Objective:

$$\min J(W)$$

Update rules:

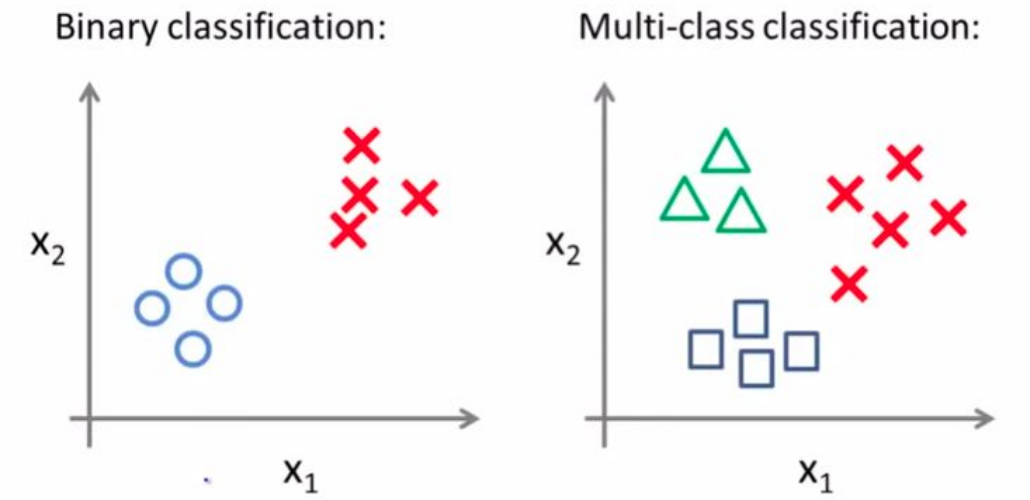
$$W_j = W_j - \alpha \frac{d}{d_{W_j}} J(W_j)$$

Derivatives:

$$\frac{d}{d_W} J(W_j) = \sum_{i=1}^{m} (Z(x^i) - y^i) * x_j^i$$

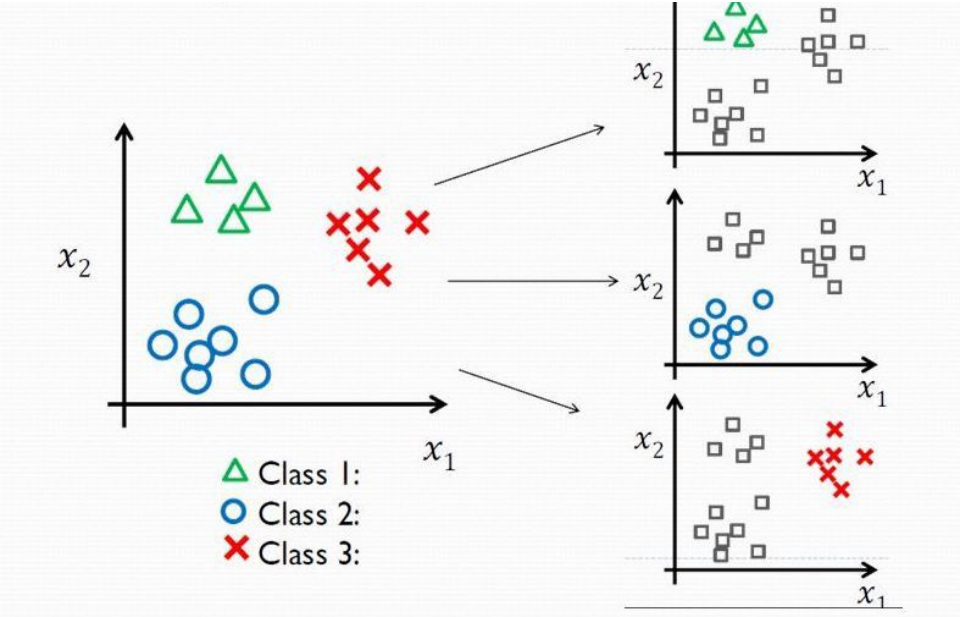
Multinomial Logistic Regression

In statistics, multinomial logistic regression is a classification method that generalizes logistic regression to multiclass problems, i.e. with more than two possible discrete outcomes.



One Vs All

Train a logistic regression classifier h(x) for each class i to predict the probability that y = i.
On a new input x, to make a prediction, pick the class i that maximizes
max h(x)



Stratified

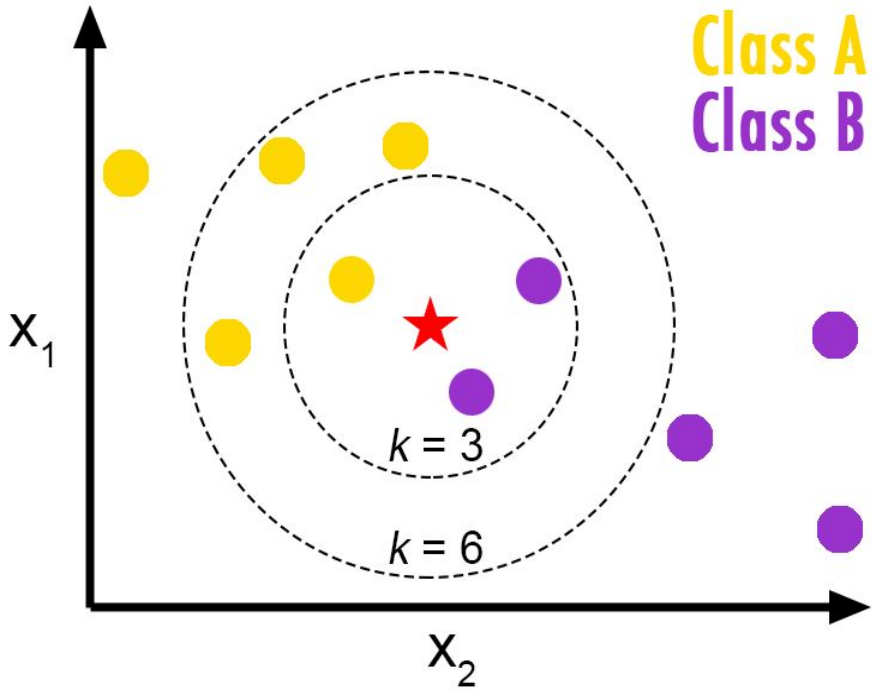
Stratified sampling aims at splitting a data set so that each split is similar with respect to something. In a classification setting, it is often chosen to ensure that the train and test sets have approximately the same percentage of samples of each target class as the complete set.

```
model_selection.train_test_split(x,y,test_size=0.1,stratify=y)
```

Code

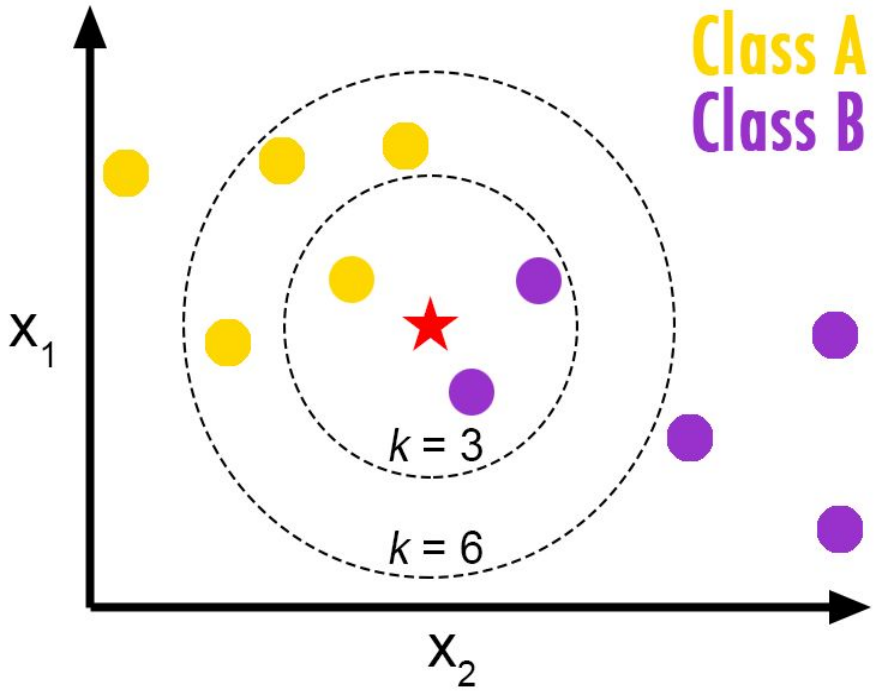
KNN Classification

KNN is a non-parametric and lazy learning algorithm. Non-parametric means there is no assumption for underlying data distribution. In other words, the model structure determined from the dataset. This will be very helpful in practice where most of the real world datasets do not follow mathematical theoretical assumptions.



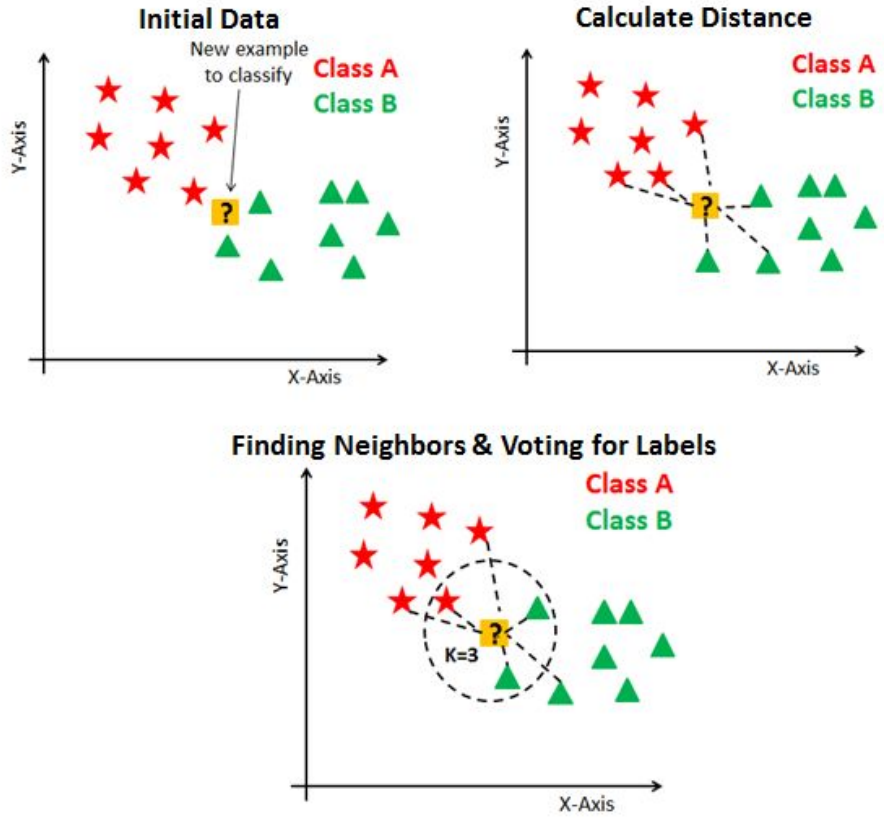
KNN Classification

Lazy algorithm means it does not need any training data points for model generation. All training data used in the testing phase. This makes training faster and testing phase slower and costlier. Costly testing phase means time and memory. In the worst case, KNN needs more time to scan all data points and scanning all data points will require more memory for storing training data.



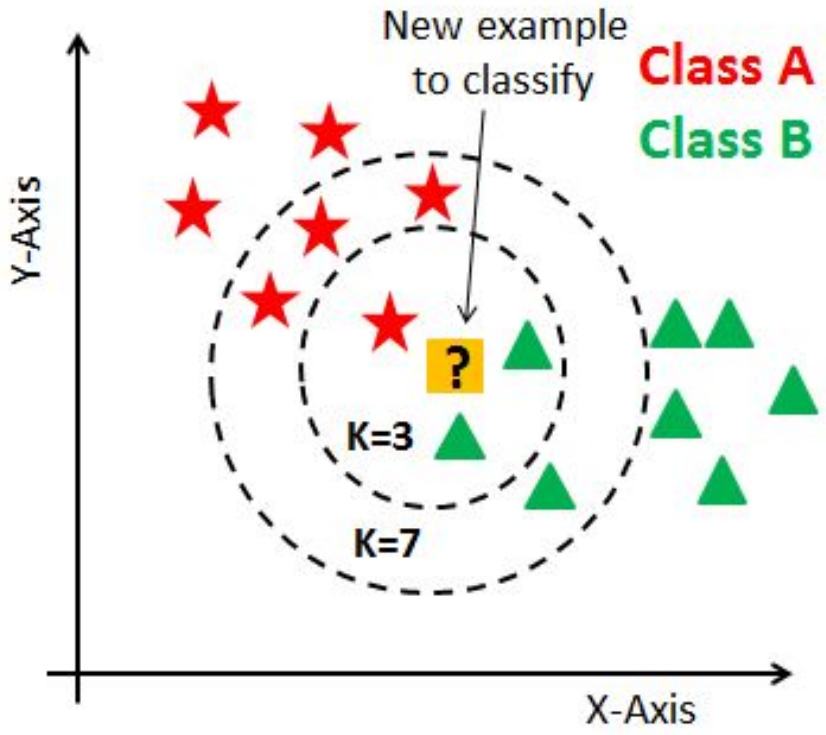
How does the KNN algorithm work?

1- Calculate distance
2- Find closest neighbors
3- Vote for labels
4-Take the majority Vote



the number of neighbors in KNN?

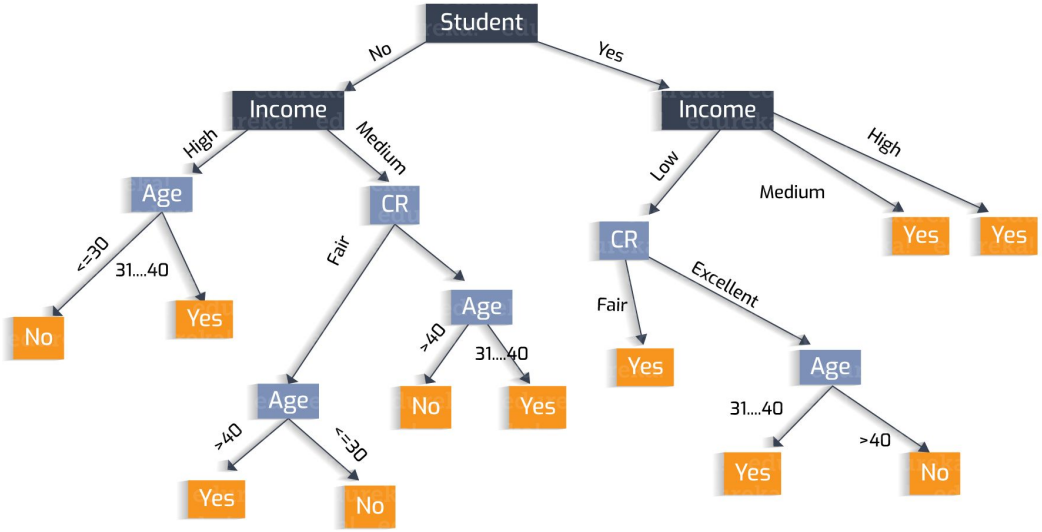
the question arises that How to choose the optimal number of neighbors?



Code time

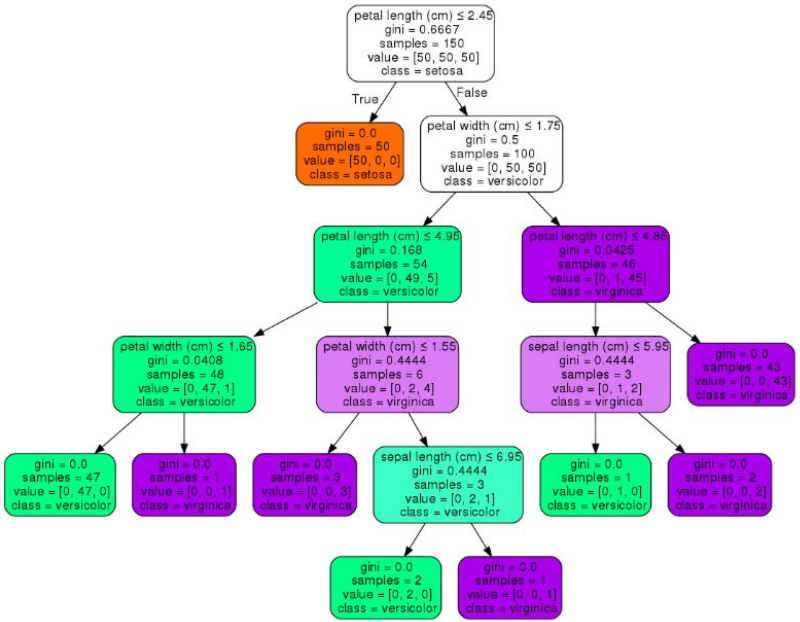
Decision Tree

A decision tree is a flowchart-like structure in which each internal node represents a "test" on an attribute (e.g. whether a coin flip comes up heads or tails), each branch represents the outcome of the test, and each leaf node represents a class label (decision taken after computing all attributes)

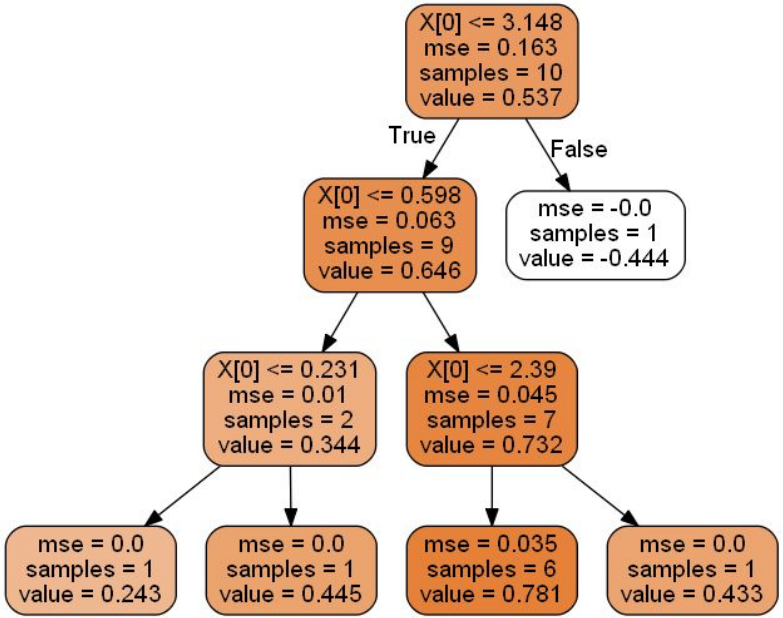


Decision Tree

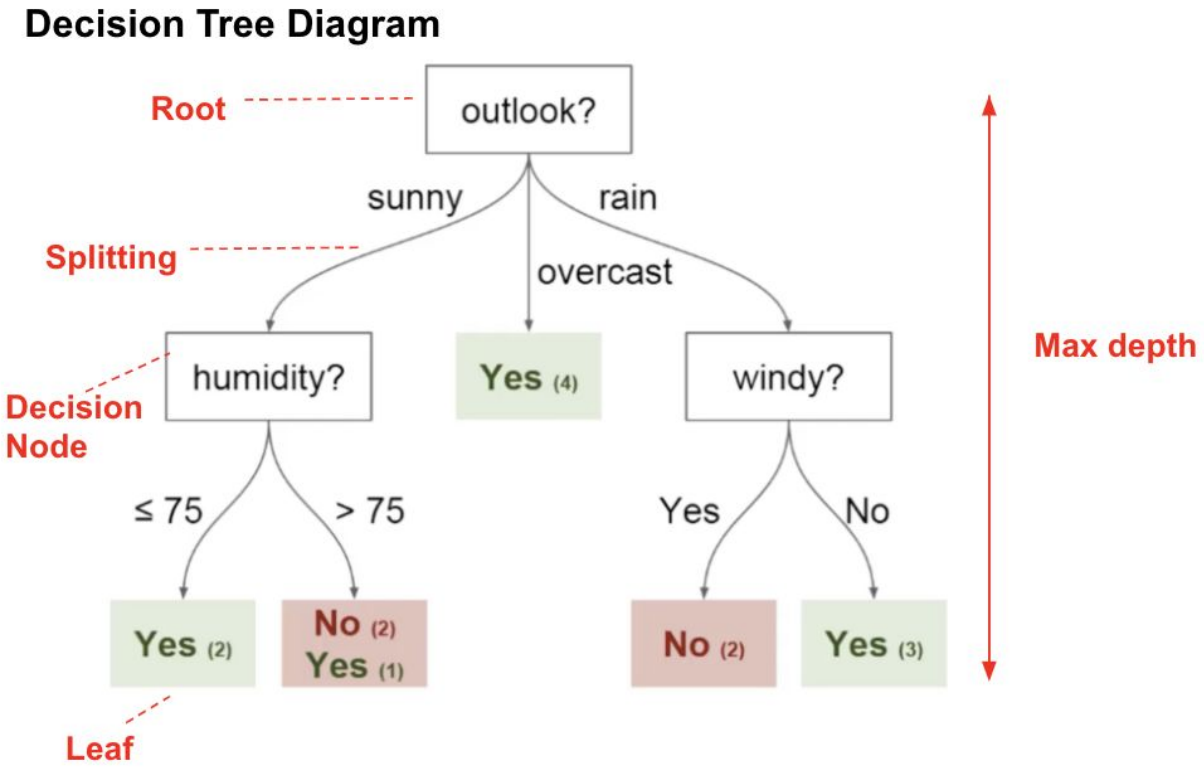
1. Categorical variable decision tree



2.Continuous Variable Decision Tree



Important Terminology related to Decision Trees



Important Terminology related to Decision Trees

1.Root Node: It represents the entire population or sample and this further gets divided into two or more homogeneous sets.

2.Splitting: It is a process of dividing a node into two or more sub-nodes.

3.Decision Node: When a sub-node splits into further sub-nodes, then it is called the decision node.
4.Leaf / Terminal Node: Nodes do not split is called Leaf or Terminal node.

5.Pruning: When we remove sub-nodes of a decision node, this process is called pruning. You can say the opposite process of splitting.

6.Branch / Sub-Tree: A subsection of the entire tree is called branch or sub-tree.

7.Parent and Child Node: A node, which is divided into sub-nodes is called a parent node of sub-nodes whereas sub-nodes are the child of a parent node.

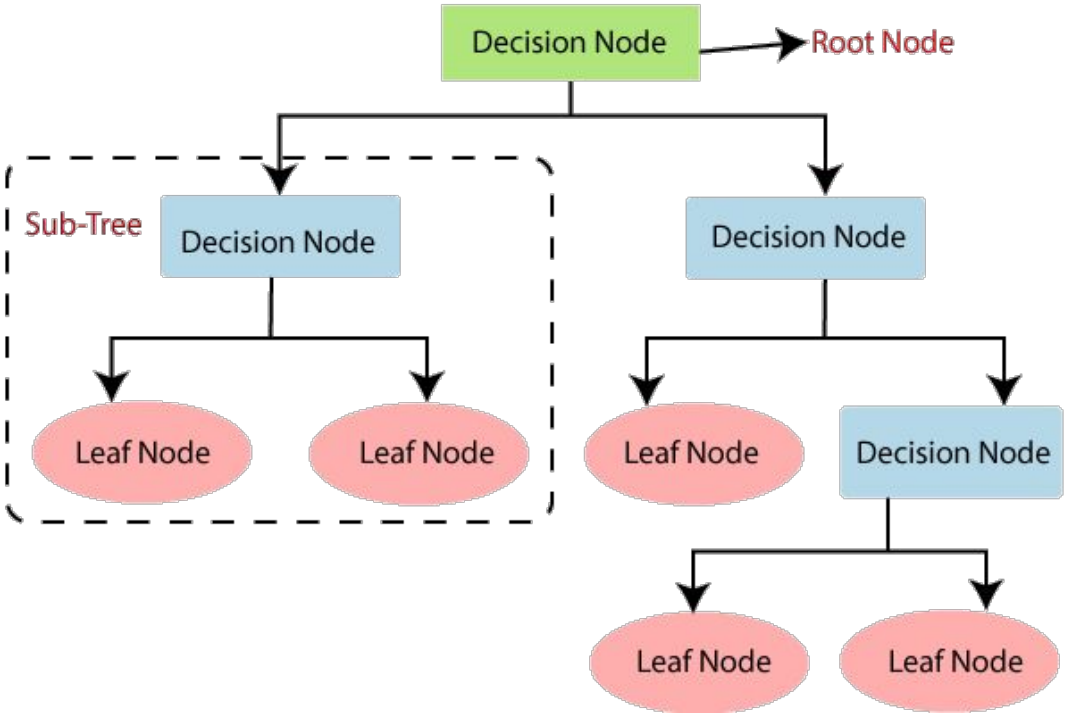
Assumptions while creating Decision Tree

- In the beginning, the whole training set is considered as the **root.**
- Feature values are preferred to be categorical. If the values are continuous then they are discretized prior to building the model.
- Records are **distributed recursively** on the basis of attribute values.
- Order to placing attributes as root or internal node of the tree is done by using some statistical approach.

How do Decision Trees work?

The decision of making strategic splits heavily affects a tree's accuracy. The decision criteria are different for classification and regression trees.

Decision trees use multiple algorithms to decide to split a node into two or more sub-nodes. The creation of sub-nodes increases the homogeneity of resultant sub-nodes. In other words, we can say that the purity of the node increases with respect to the target variable. The decision tree splits the nodes on all available variables and then selects the split which results in most homogeneous sub-nodes.



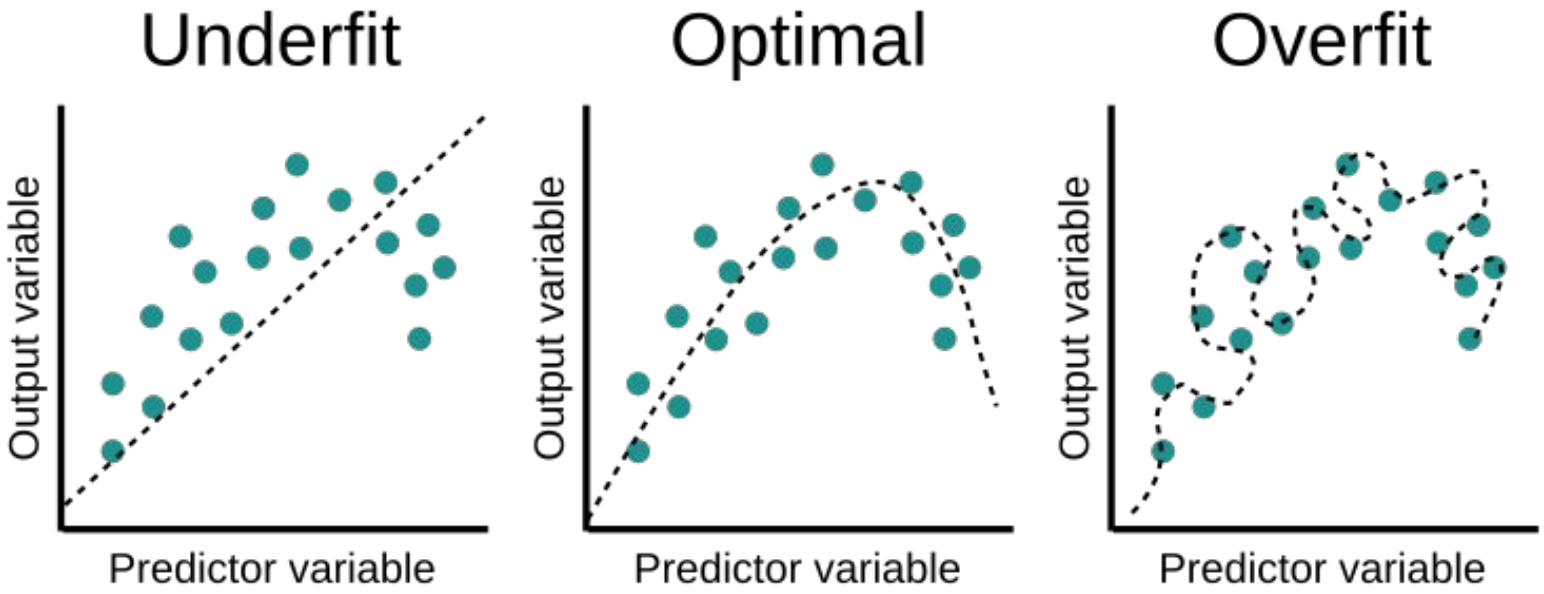
Code

accuracy score

$$\texttt{accuracy}(y, \hat{y}) = \frac{1}{n_{\text{samples}}} \sum_{i=0}^{n_{\text{samples}}-1} 1(\hat{y}_i = y_i)$$

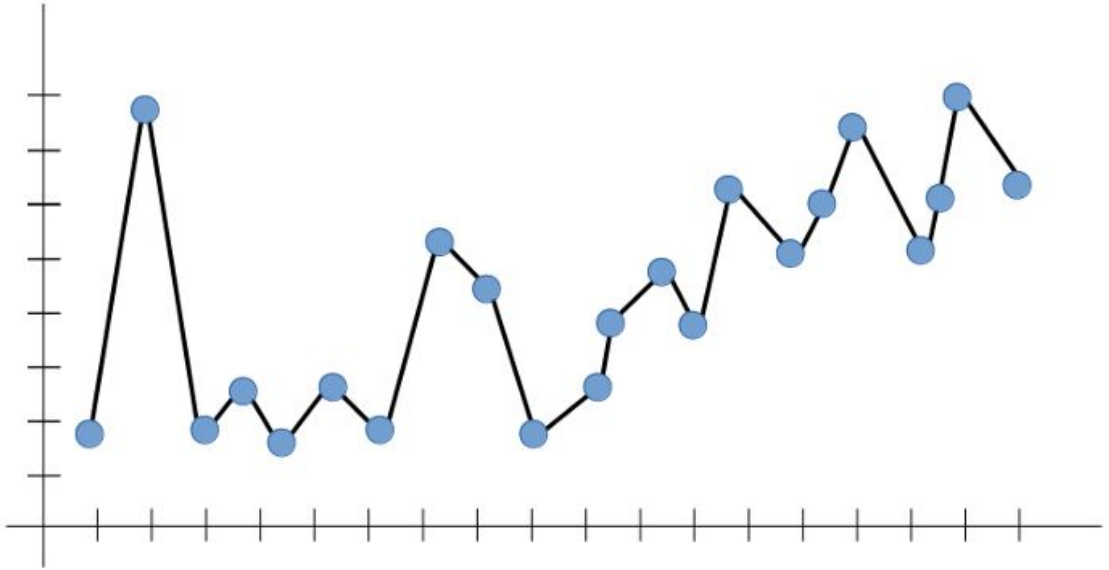
sklearn.metrics.accuracy_score(y,pred)

overfitting and underfitting



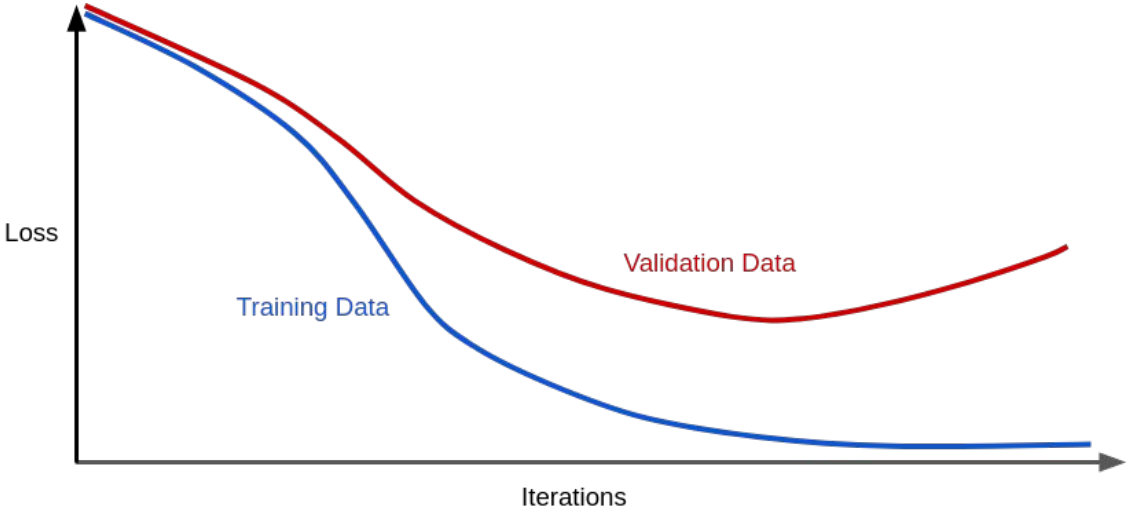
Overfitting

When we run our training algorithm on the data set, we allow the overall cost (i.e. distance from each point to the line) to become smaller with more iterations. Leaving this training algorithm run for long leads to minimal overall cost. However, this means that the line will be fit into all the points (including noise), catching secondary patterns that may not be needed for the generalizability of the model.



l2 regularization

L2 relles on assumption that model with small weights is simpler than a model with larger weights .thus by penalizing the square values of the weights in cost function you will make all weights to smaller values. It becomes too costly for the cost to have larger weights this leads to a smooth model in which the output change mode slowly as input change



$$\text{minimize}(\text{Loss}(\text{Data}|\text{Model}) + \text{complexity}(\text{Model}))$$

$$J_{regularized} = \underbrace{-\frac{1}{m}\sum_{i=1}^{m}\left(y^{(i)}\log\left(a^{[L](i)}\right) + (1-y^{(i)})\log\left(1-a^{[L](i)}\right)\right)}_{\text{cross-entropy cost}} + \underbrace{\frac{1}{m}\frac{\lambda}{2}\sum_{l}\sum_{k}\sum_{j}W_{k,j}^{[l]2}}_{\text{L2 regularization cost}}$$

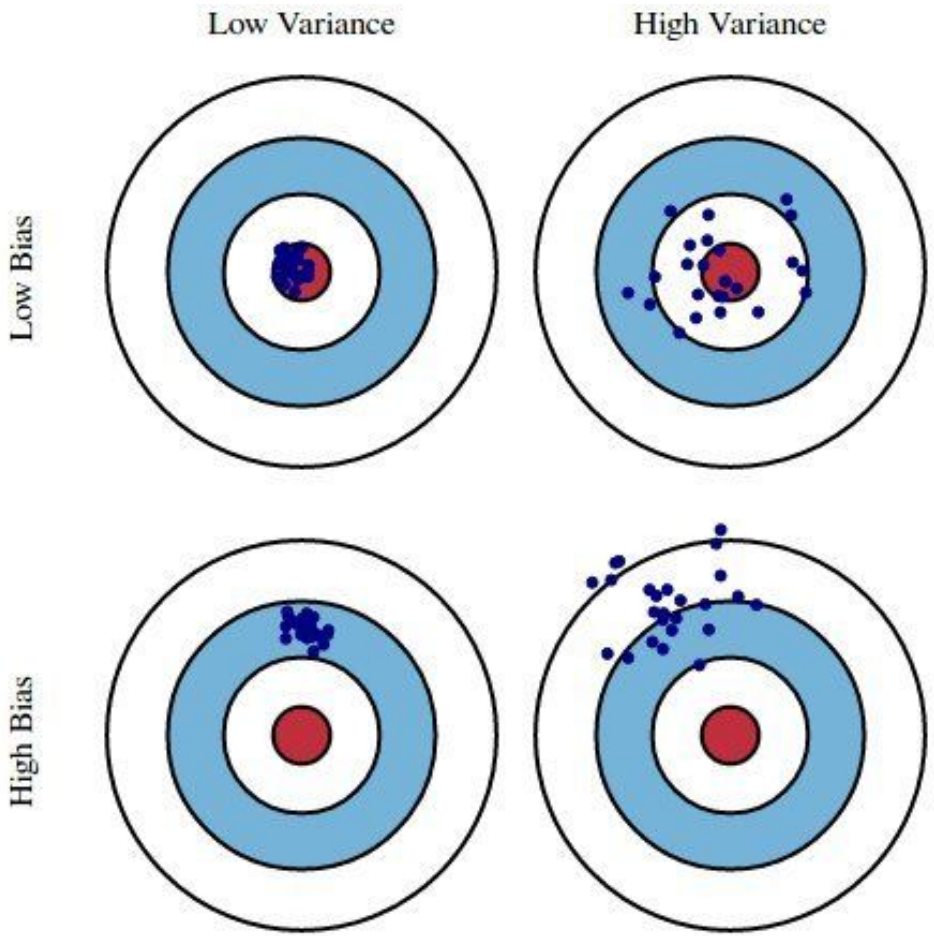
$$L_2 \text{ regularization term} = ||\boldsymbol{w}||_2^2 = w_1^2 + w_2^2 + \ldots + w_n^2$$

Evaluation Metrics

Regression

What is Variance?

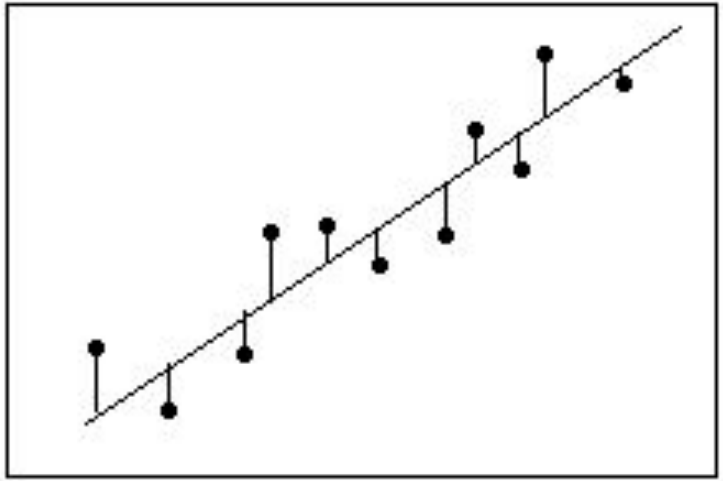
In terms of linear regression, variance is a measure of how far observed values differ from the average of predicted values, i.e., their difference from the predicted value mean. The goal is to have a value that is low. What low means is quantified by the r2 score



r2 score

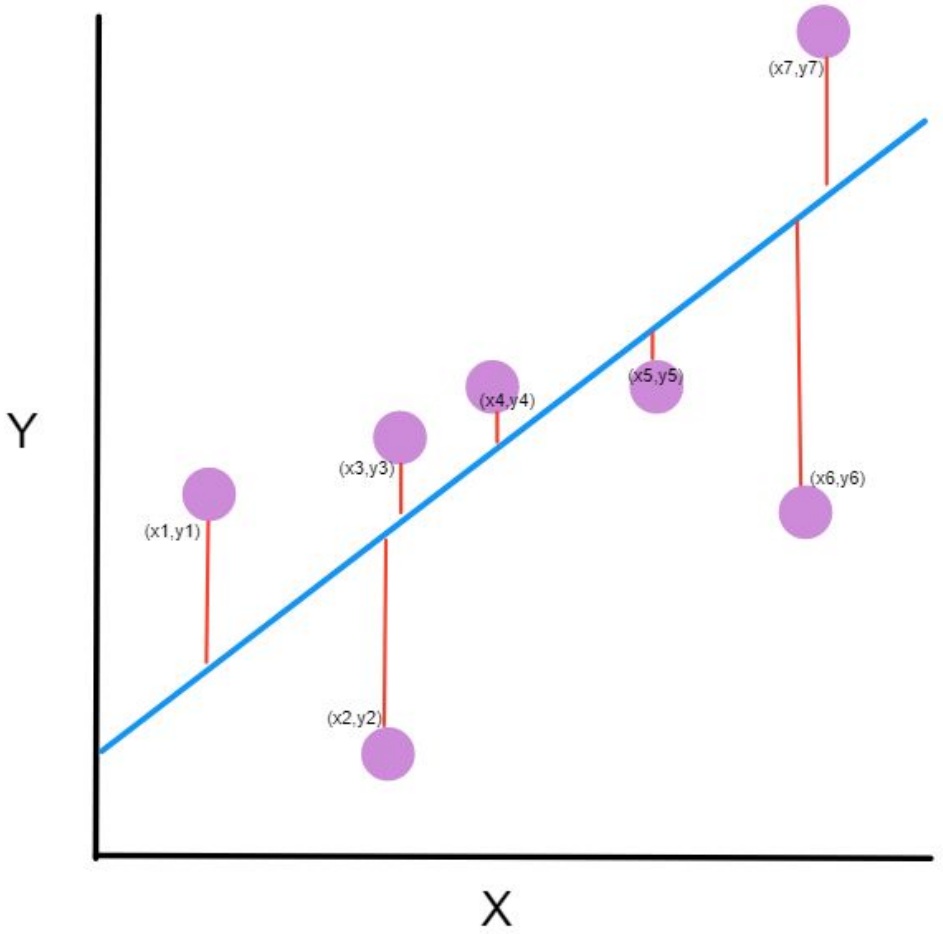
the r2 score varies between 0 and 100%. It is closely related to the MSE .

if it is 100%, the two variables are perfectly correlated, i.e., with no variance at all. A low value would show a low level of correlation, meaning a regression model that is not valid, but not in all cases.



Mean Square Error (MSE)

Mean square error (MSE) is the average of the square of the errors. The larger the number the larger the error. Error in this case means the difference between the observed values y1, y2, y3, … and the predicted ones pred(y1), pred(y2), pred(y3), … We square each difference (pred(yn) – yn)) ** 2 so that negative and positive values do not cancel each other out.



Classification

True Positive (TP)

The predicted value matches the actual value
The actual value was positive and the model predicted a positive value

True Negative (TN)

The predicted value matches the actual value
The actual value was negative and the model predicted a negative value

		Predicted	
		Negative	**Positive**
Actual	**Negative**	True Negative	False Positive
	Positive	False Negative	True Positive

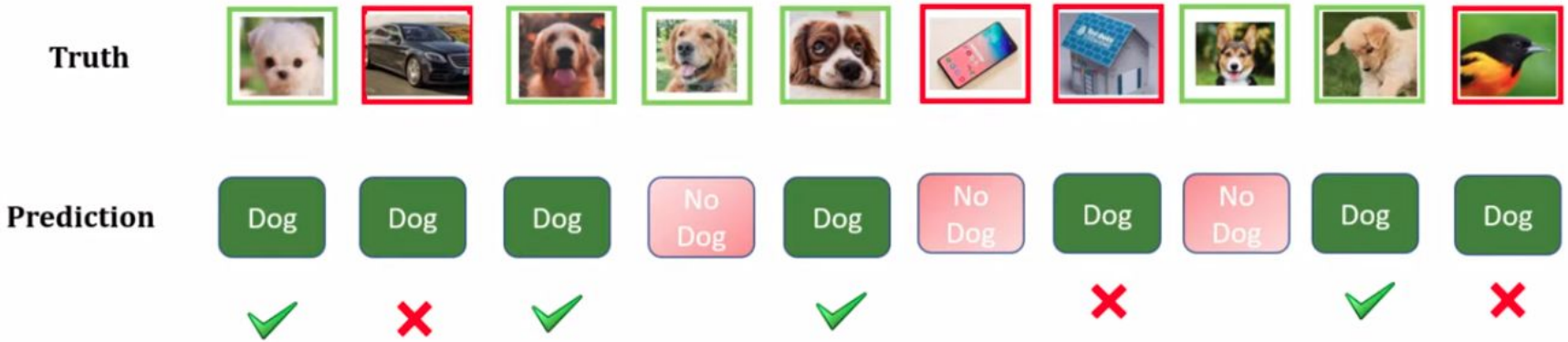
False Positive (FP) – Type 1 error

The predicted value was falsely predicted
The actual value was negative but the model predicted a positive value
Also known as the Type 1 error

False Negative (FN) – Type 2 error

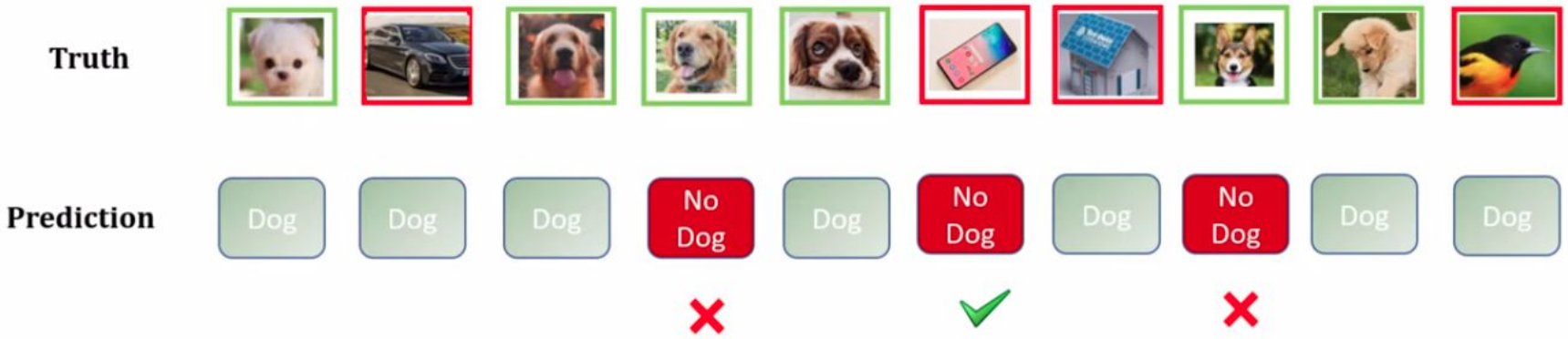
The predicted value was falsely predicted
The actual value was positive but the model predicted a negative value
Also known as the Type 2 error

		Predicted	
		Negative	**Positive**
Actual	**Negative**	True Negative	False Positive
	Positive	False Negative	True Positive



True Positive = 4

False Positive = 3

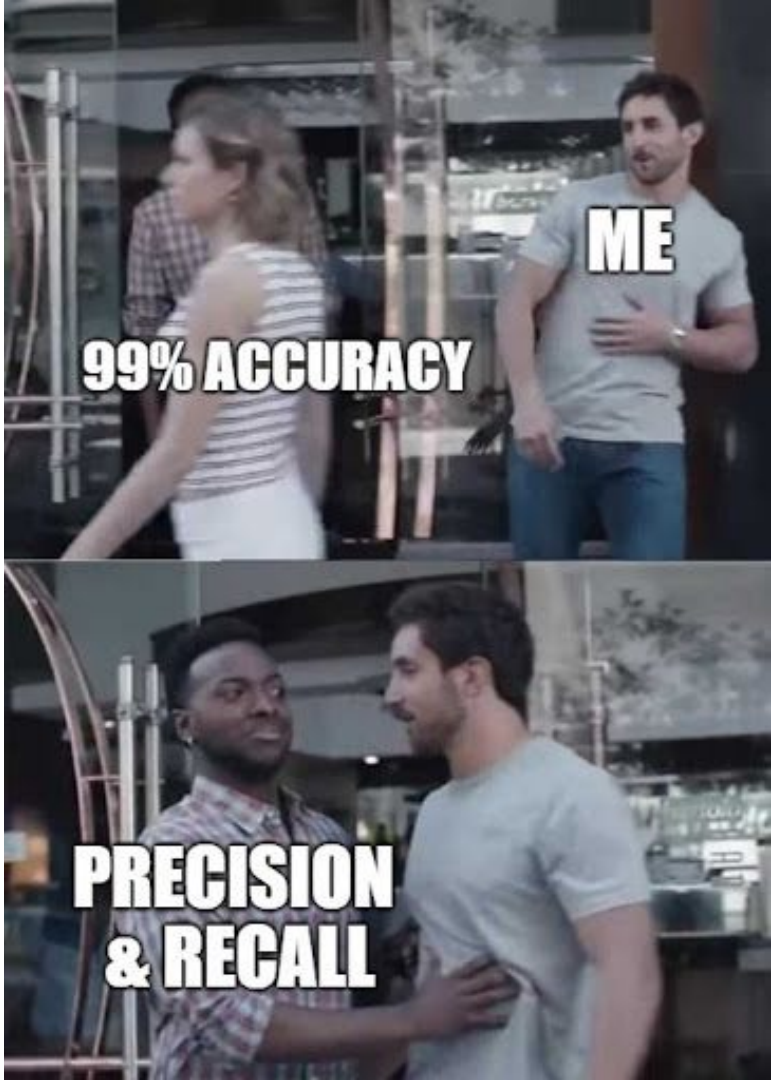


True Negative = 1

False Negative = 2

How many we got right? → 5

Accuracy → 5/10 → **0.5**


ME
99% ACCURACY
PRECISION
& RECALL

Precision and Recall

Precision is a good measure to determine, when the costs of False Positive is high. For instance, email spam detection. In email spam detection, a false positive means that an email that is non-spam (actual negative) has been identified as spam (predicted spam). The email user might lose important emails if the precision is not high for the spam detection model.

		Predicted	
		Negative	**Positive**
Actual	**Negative**	True Negative	False Positive
	Positive	False Negative	True Positive

$$\text{Precision} = \frac{TP}{TP + FP}$$

Precision and Recall

Recall shall be the model metric we use to select our best model when there is a high cost associated with False Negative. in sick patient detection. If a sick patient (Actual Positive) goes through the test and predicted as not sick (Predicted Negative). The cost associated with False Negative will be extremely high if the sickness is contagious.

		Predicted	
		Negative	**Positive**
Actual	**Negative**	True Negative	False Positive
	Positive	False Negative	True Positive

$$\text{Recall} = \frac{TP}{TP + FN}$$

Truth										
Prediction	Dog	Dog	Dog	No Dog	Dog	No Dog	Dog	No Dog	Dog	Dog
	✓	✗	✓		✓		✗		✓	✗

True Positive = 4

False Positive = 3

Precision is out of all dog predictions how many you got it right?

Precision = 4 / 7 = 0.57

Precision = TP / (TP + FP)

Recall is out of all dog truth how many you got it right?

False Negative = 2

True Positive = 4

Recall = 4 / 6 = 0.67

Recall = TP / (TP + FN)

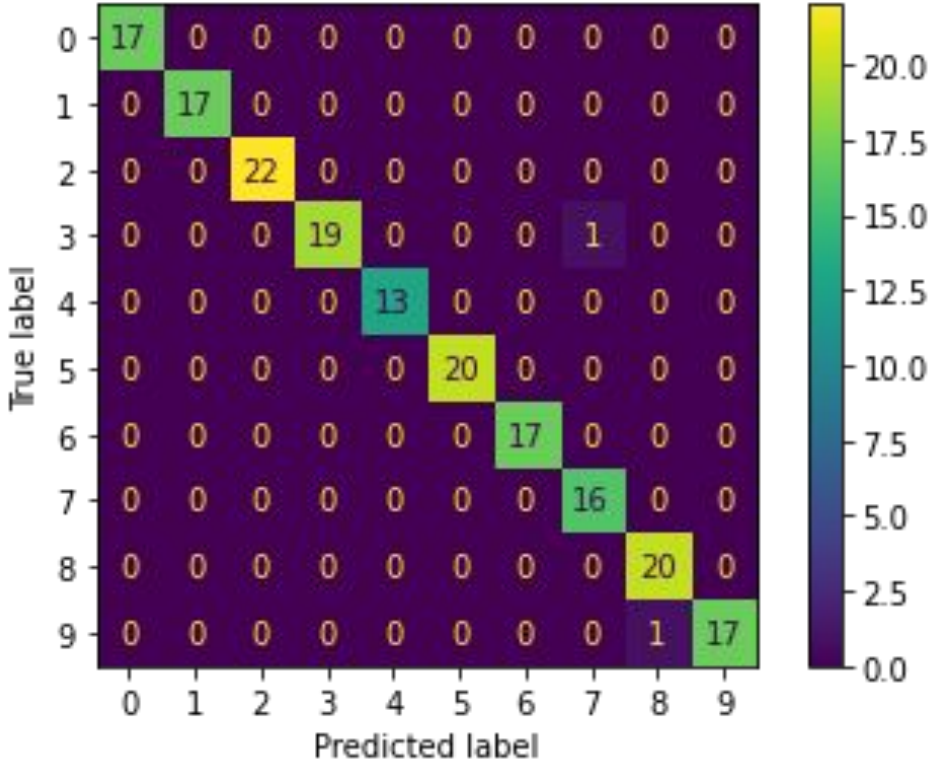
F1 score

F1 Score might be a better measure to use if we need to seek a balance between Precision and Recall AND there is an uneven class distribution (large number of Actual Negatives).

$$F1 = 2 \times \frac{Precision * Recall}{Precision + Recall}$$

confusion matrix

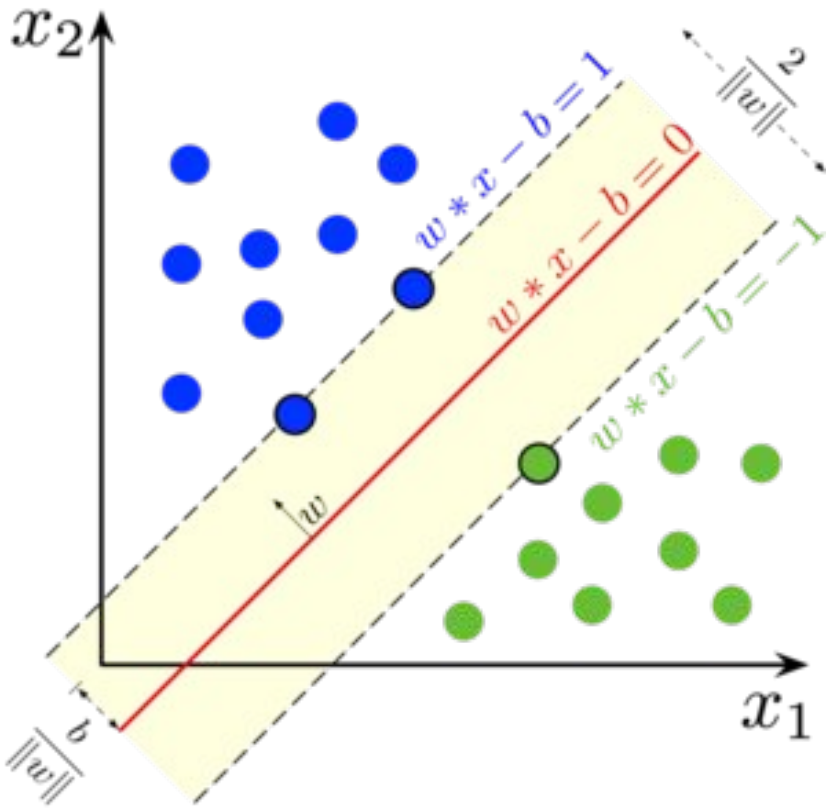
A Confusion matrix is an N x N matrix used for evaluating the performance of a classification model, where N is the number of target classes. The matrix compares the actual target values with those predicted by the machine learning model. This gives us a holistic view of how well our classification model is performing and what kinds of errors it is making.



Support Vector Machine

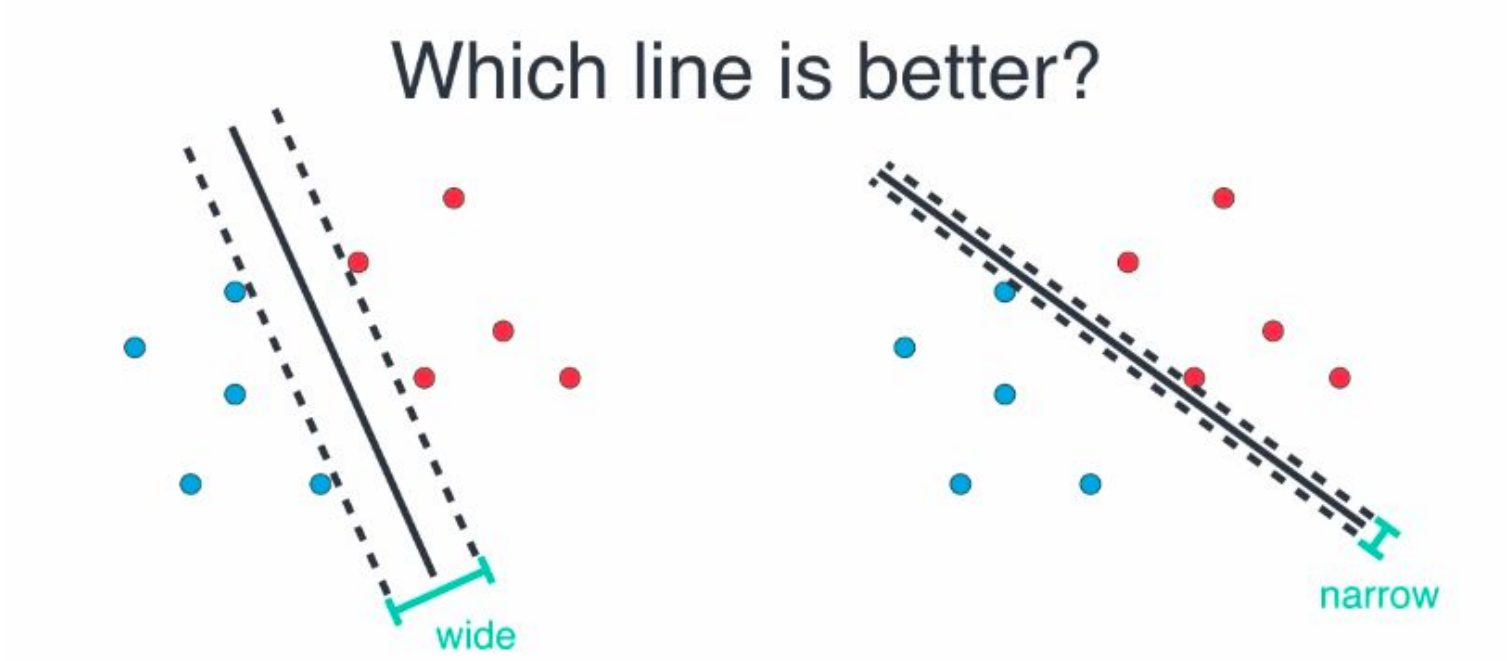
Support Vector Machine

A support vector machine (SVM) is a supervised machine learning model that uses classification algorithms for two-group classification problems. After giving an SVM model sets of labeled training data for each category, they're able to categorize it.

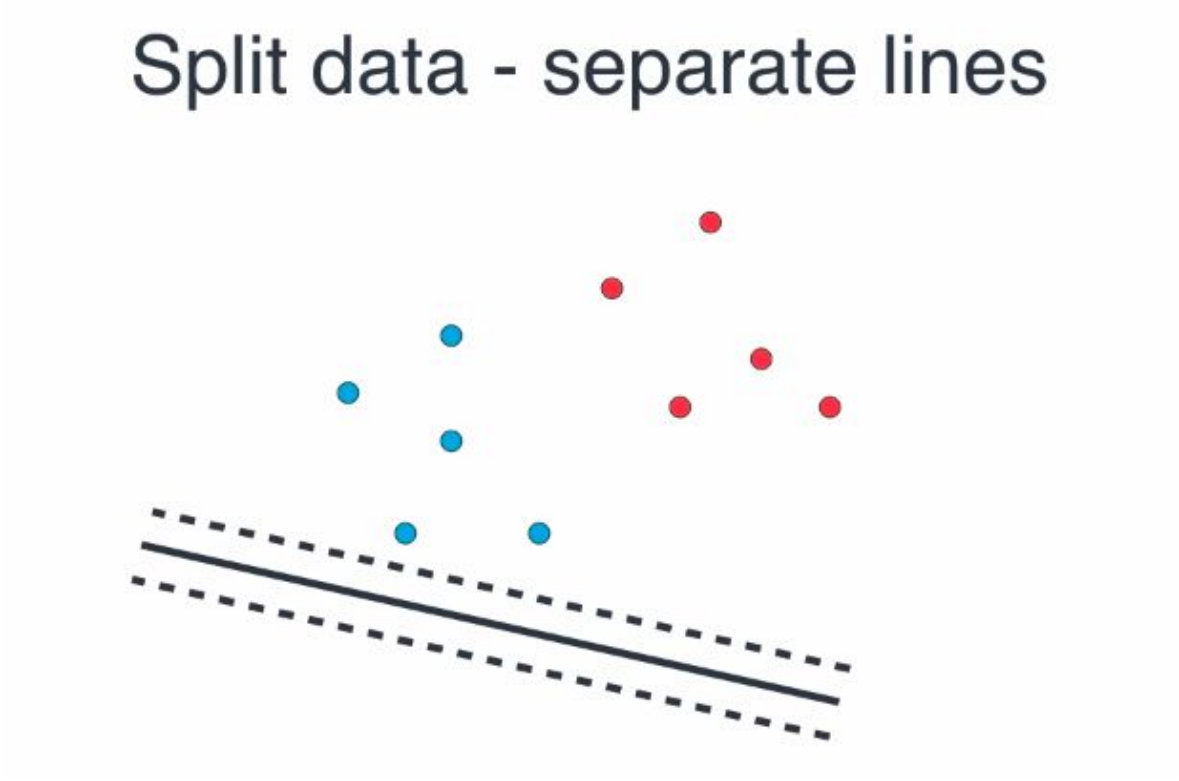


Support Vector Machine

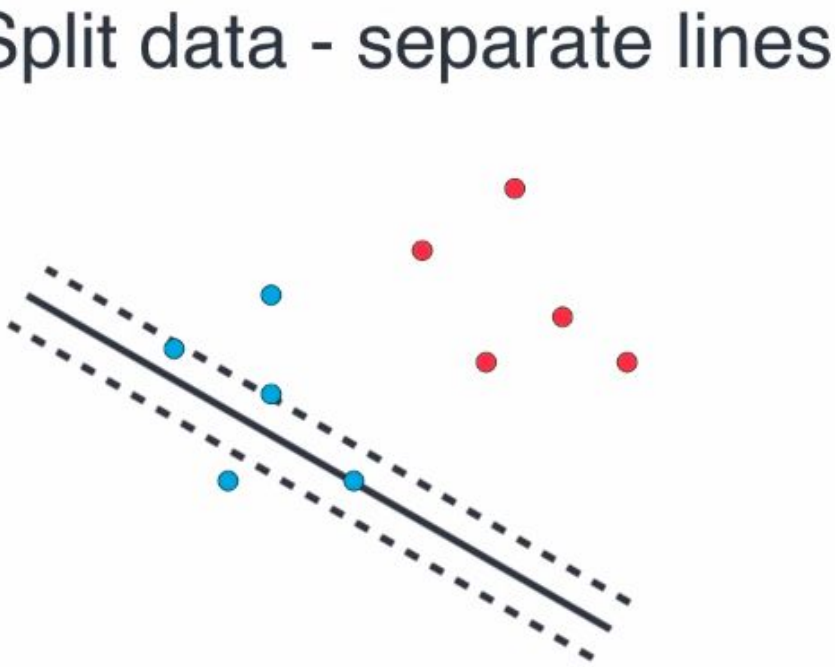
Support Vector Machine



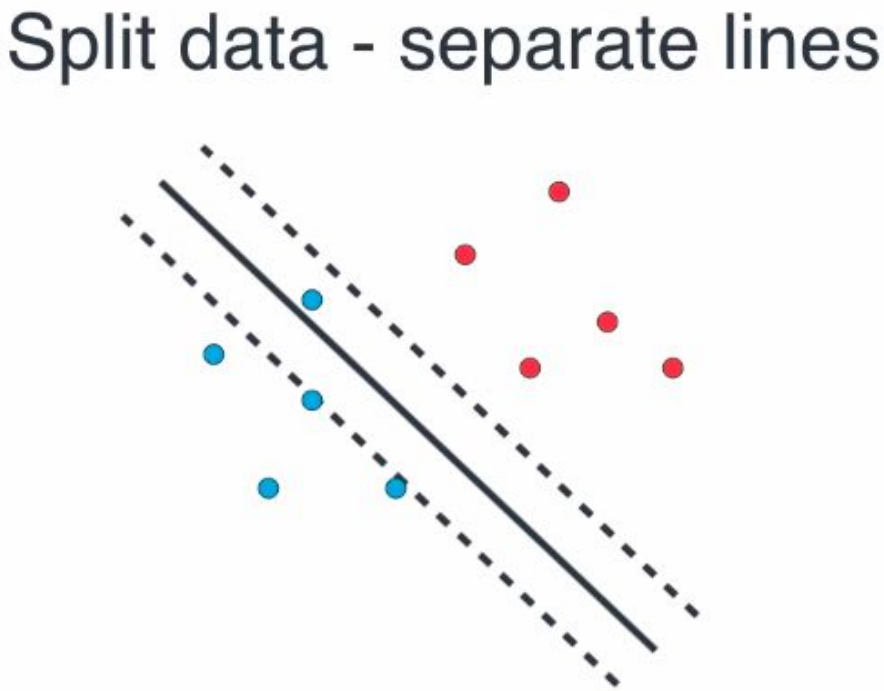
Support Vector Machine



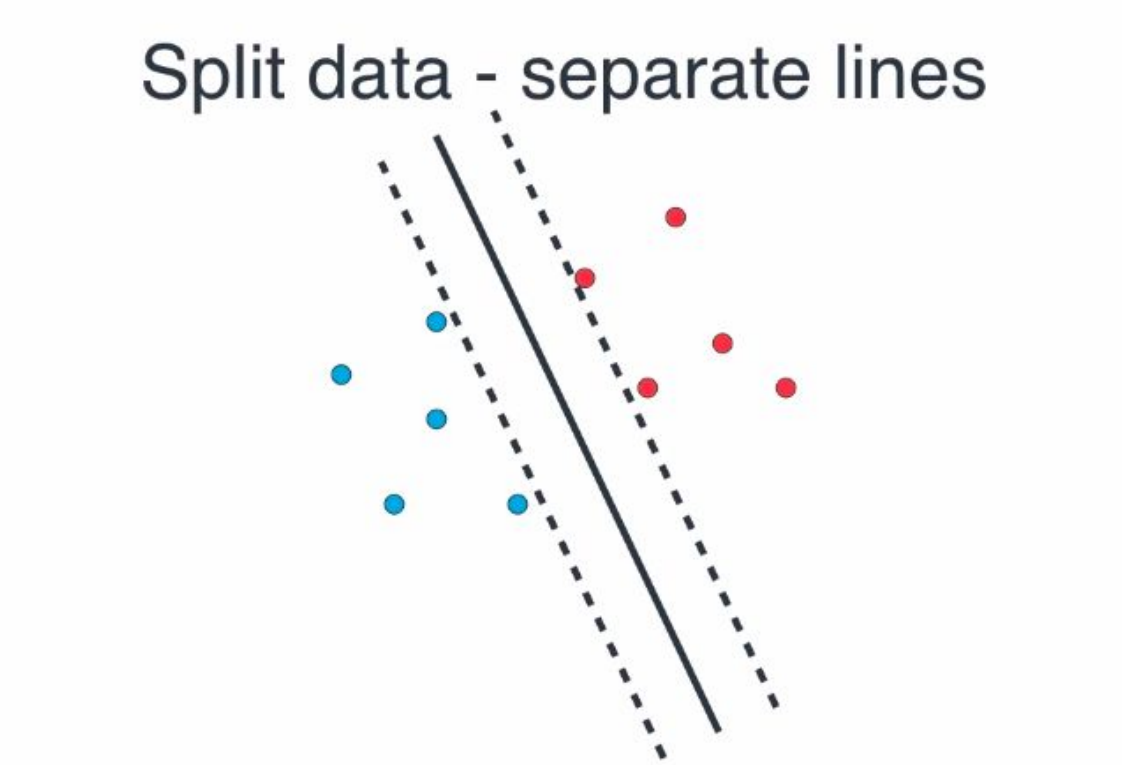
Support Vector Machine



Support Vector Machine



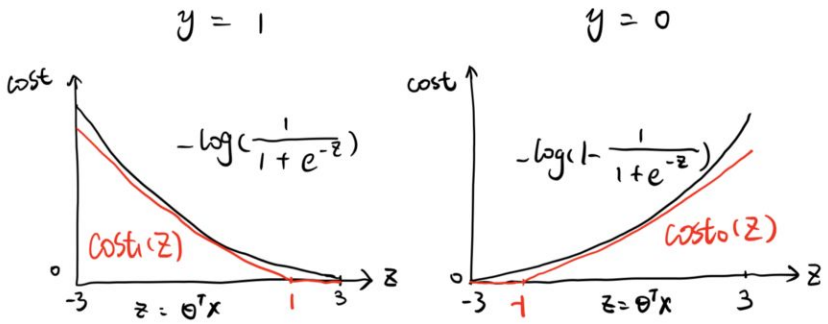
Support Vector Machine



hinge loss

In machine learning, the hinge loss is a loss function used for training classifiers. The hinge loss is used for "maximum-margin" classification, most notably for support vector machines (SVMs).

$$J(W) = -\frac{1}{m}(\sum_{i=1}^{m} y^i log(Z(x^i))) + (1 - y^i)(1 - log(Z(x^i))))$$

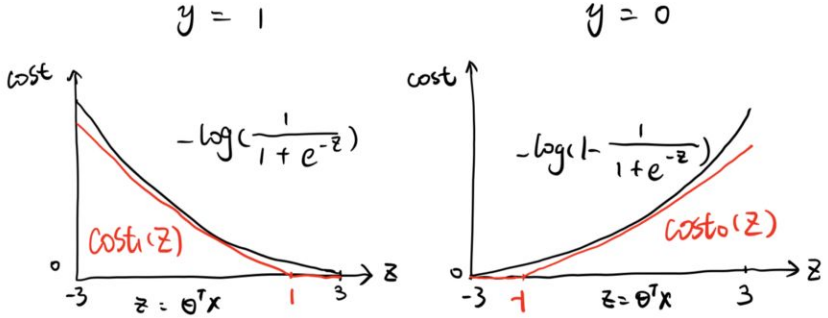


SVM Hypothesis:

$$h_\theta(x) = \begin{cases} 1 & \text{if } \theta^T x >= 0 \\ 0 & \text{otherwise} \end{cases}$$

hinge loss

In machine learning, the hinge loss is a loss function used for training classifiers. The hinge loss is used for "maximum-margin" classification, most notably for support vector machines (SVMs).



$$Cost(h_\theta(x), y) = \begin{cases} max(0, 1 - \theta^T x) & \text{if } y = 1 \\ max(0, 1 + \theta^T x) & \text{if } y = 0 \end{cases}$$

$$J(\theta) = \sum_{i=1}^{m} y^{(i)} Cost_1(\theta^T(x^{(i)}) + (1 - y^{(i)}) Cost_0(\theta^T(x^{(i)})$$

$$J(\theta) = \sum_{i=1}^{m} y^{(i)} max(0, 1 - \theta^T x) + (1 - y^{(i)}) max(0, 1 + \theta^T x)$$

$$m = number\ of\ samples$$

$$J(\theta) = C[\sum_{i=1}^{m} y^{(i)} Cost_1(\theta^T(x^{(i)}) + (1 - y^{(i)}) Cost_0(\theta^T(x^{(i)})] + \frac{1}{2} \sum_{j=1}^{n} \theta_j^2$$

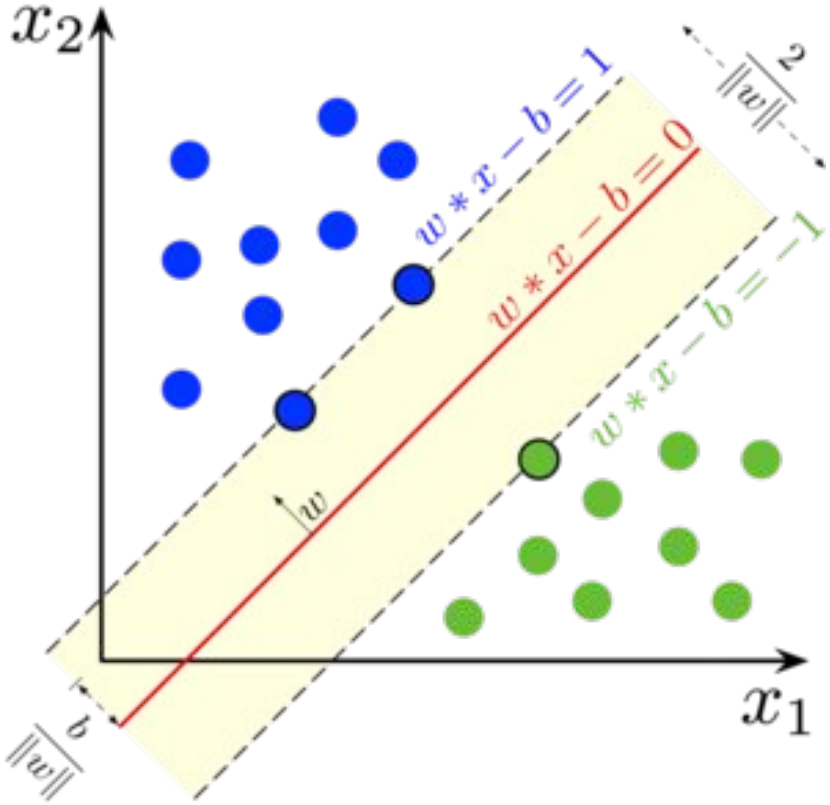
$$m = number\ of\ samples, \quad n = number\ of\ features$$

Margin


do you want to enforce social distancing?
yes, please
use SVM to maximize the margin

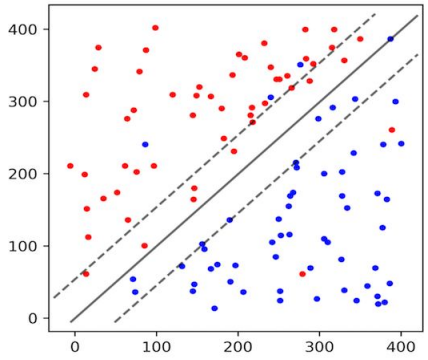
Margin

Margin is the perpendicular distance between the closest data points and Hyperplane (on both sides),The best optimised line(hyperplane)with maximum margin is termed as margin maximal hyperplan.
The closest points where the margin distance is calculated are considered as the support vectors.

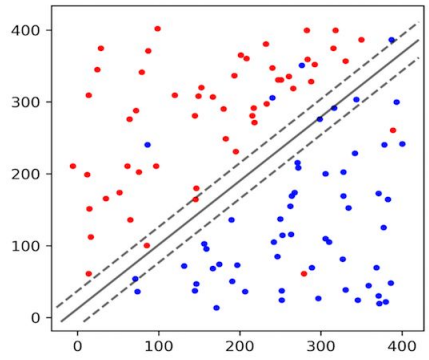


C

SVM still finds the best hyperplane by solving an optimization problem that tries to increase the distance of the hyperplane from the two classes while trying to make sure many training examples are classified properly.
This tradeoff is controlled by a parameter called C. When the value of C is small, a large margin hyperplane is chosen at the expense of a greater number of misclassifications.
Conversely, when C is large, a smaller margin hyperplane is chosen that tries to classify many more examples correctly.



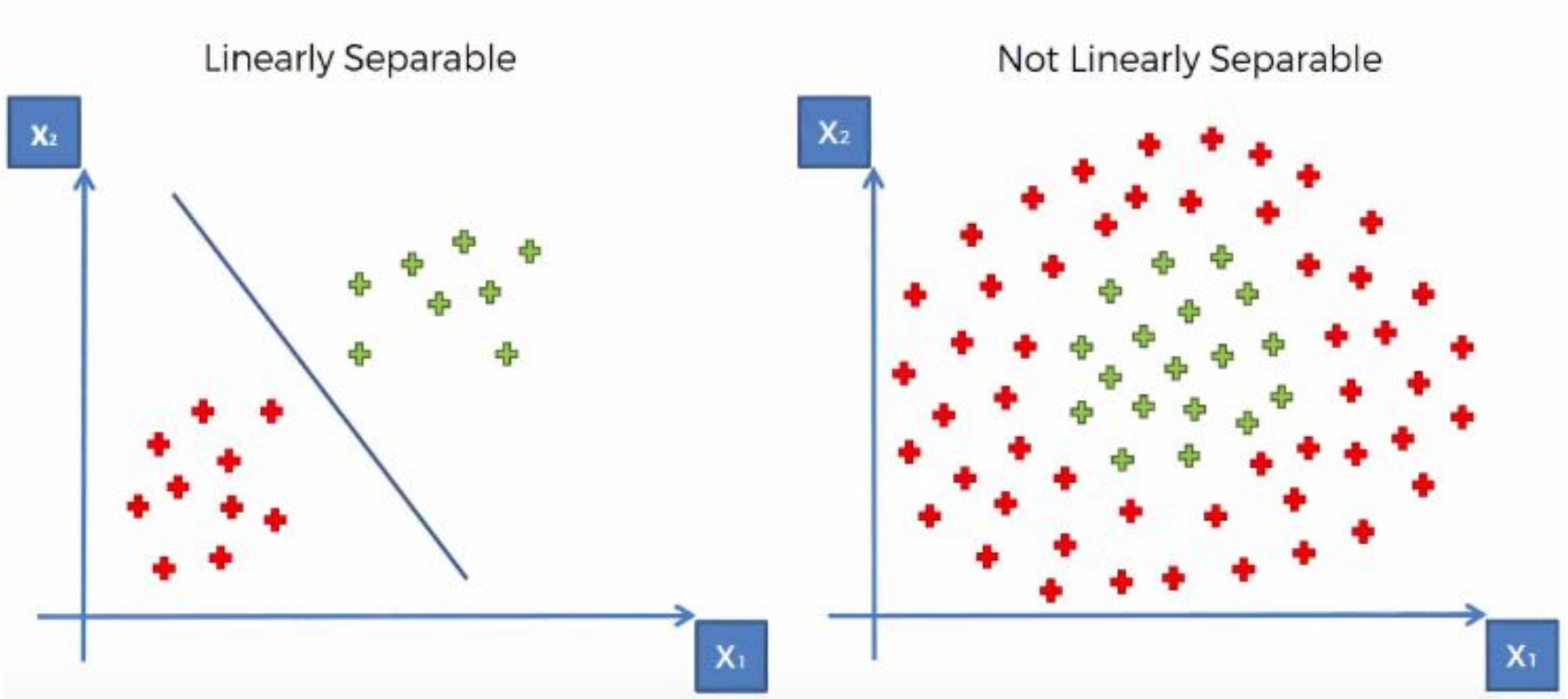
C = 1



C = 100

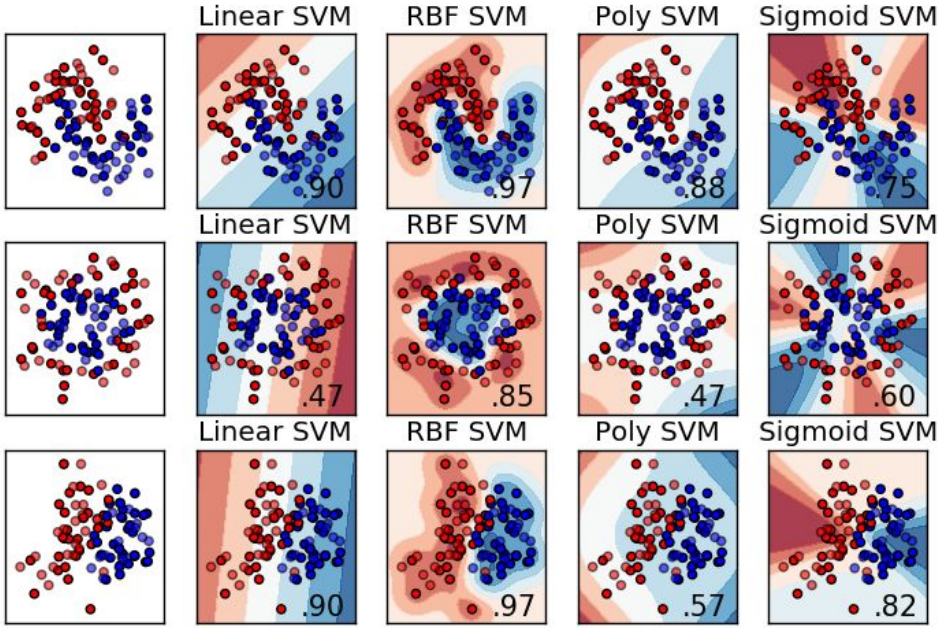
$$J(\theta) = C[\sum_{i=1}^{m} y^{(i)} Cost_1(\theta^T(x^{(i)}) + (1 - y^{(i)}) Cost_0(\theta^T(x^{(i)})] + \frac{1}{2}\sum_{j=1}^{n} \theta_j^2$$

$m = number\ of\ samples, \quad n = number\ of\ features$


Linearly Separable
X2
X1
Not Linearly Separable
X2
X1

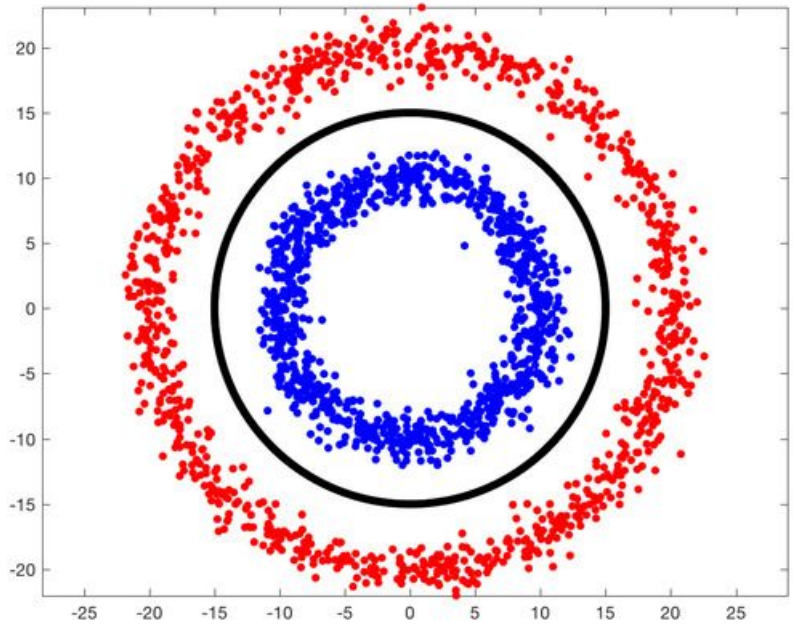
Kernels

SVM algorithms use a set of mathematical functions that are defined as the kernel. The function of kernel is to take data as input and transform it into the required form. Different SVM algorithms use different types of kernel functions. These functions can be different types. For example linear, nonlinear, polynomial, radial basis function (RBF), and sigmoid.



SVM Parameter Gamma (γ)

What if the data is not separable by a hyperplane? For example, if two classes represented by the red and blue dots are not linearly separable. The decision boundary shown in black is actually circular.
In such a case, we use the Kernel Trick where we add a new dimension to existing data and if we are lucky, in the new space, the data is linearly separable. Let's look at the Kernel Trick using an example.



SVM Parameter Gamma (γ)

The next important parameter is Gamma. The gamma parameter defines how far the influence of a single training example reaches. This means that high Gamma will consider only points close to the plausible hyperplane and low Gamma will consider points at greater distance.