Python for Data Science Comprehensive Workshop

Part 04 – Machine Learning & Deep Learning Using Scikit Learn, Tensorflow & Keras

H.M. Samadhi Chathuranga Rathnayake

B.Sc(Hons).Special in Industrial Statistics (1st Class) (UOC),
B.Eng (Hons) in Software Engineering (LMU),
CLSSWB, Dip SE, Dip IT, Dip IT & E-Com, Dip B.Mgt, Dip HRM, Dip Eng

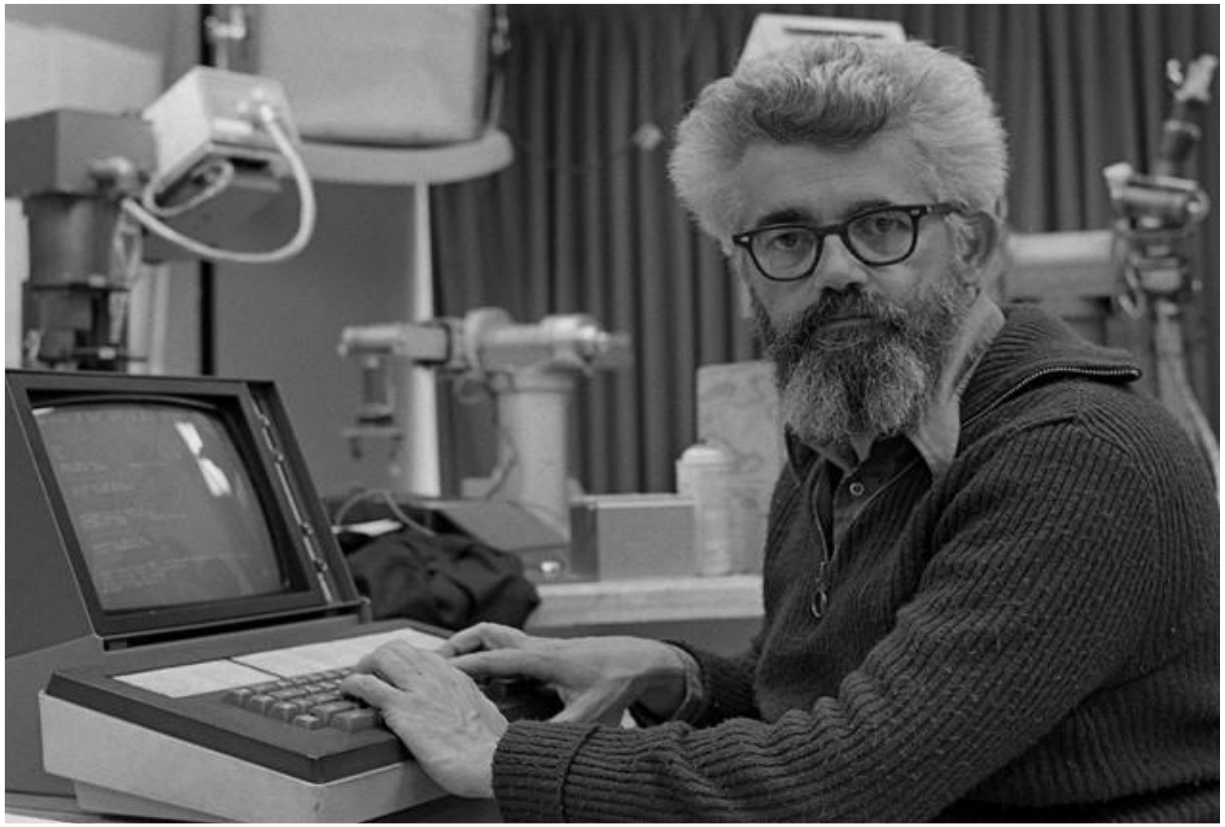
Artificial Intelligence

Artificial Intelligence is an approach to make a computer, a robot, or a product to think how smart human think AI is a study of how human brain think, learn, decide and work, when it tries to solve problems And finally this study outputs intelligent software systems

“The science and engineering of making intelligent machines, especially intelligent computer programs” –

John McCarthy

(Father of Artificial Intelligence)



Artificial Intelligence

Artificial intelligence generally false under two broad categories.

- Narrow Artificial Intelligence
- Artificial General Intelligence

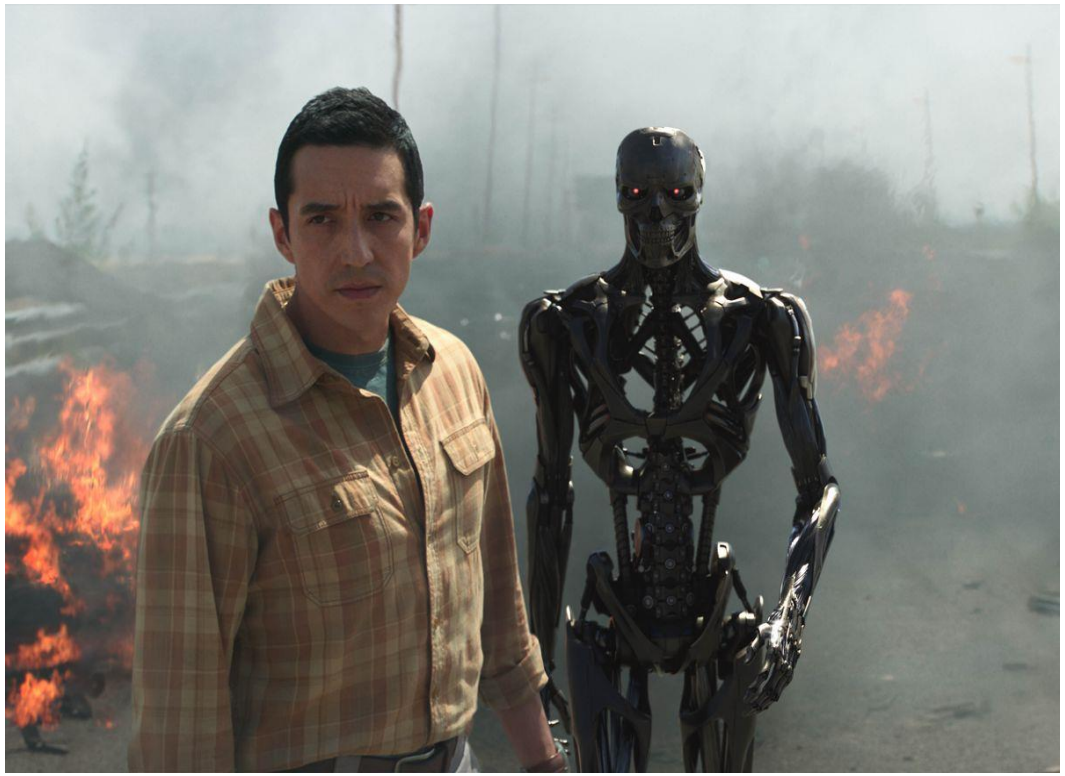
Artificial Intelligence - Narrow Artificial Intelligence

Narrow AI is all around us and is easily the most successful realization of artificial intelligence to date With its focus on performing specific tasks. Narrow AI has experienced numerous breakthroughs in the last decade that have had "significant societal benefits and have contributed to the economic vitality of the nation," according to "Preparing for the Future of Artificial Intelligence," a 2016 report released by the Obama Administration Much of Narrow AI is powered by breakthroughs in machine learning and deep learning. A few examples of Narrow AI include,

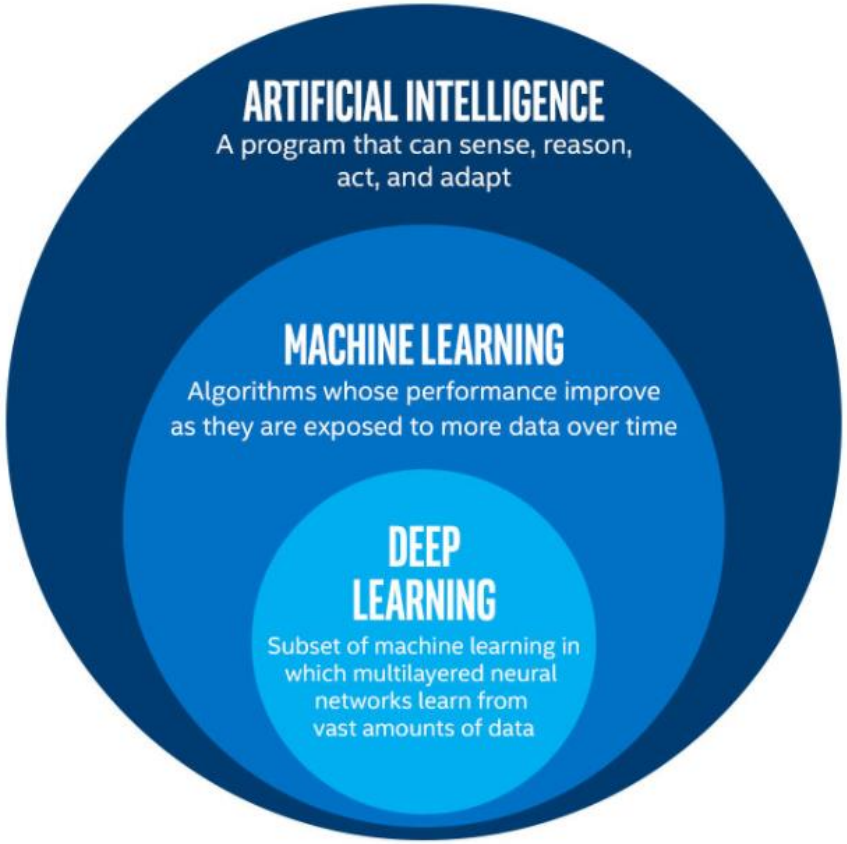
- Google search
- Image recognition software
- Siri, Alexa and other personal assistants
- Self driving cars
- IBM's Watson

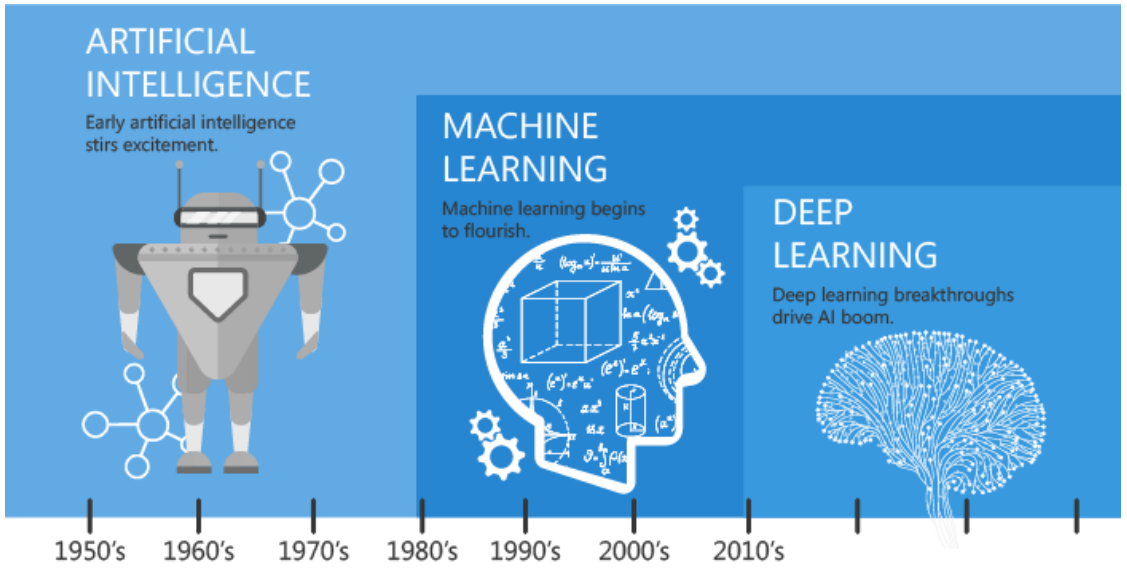
Artificial Intelligence - Artificial General Intelligence

The creation of a machine with human level intelligence that can be applied to any task is the Holy Grail for many AI researchers, but the quest for AGI has been fraught with difficulty. AGI, sometimes referred to as "Strong AI," is the kind of artificial intelligence we see in the movies, like the robots from Westworld or Data from Star Trek The Next Generation.

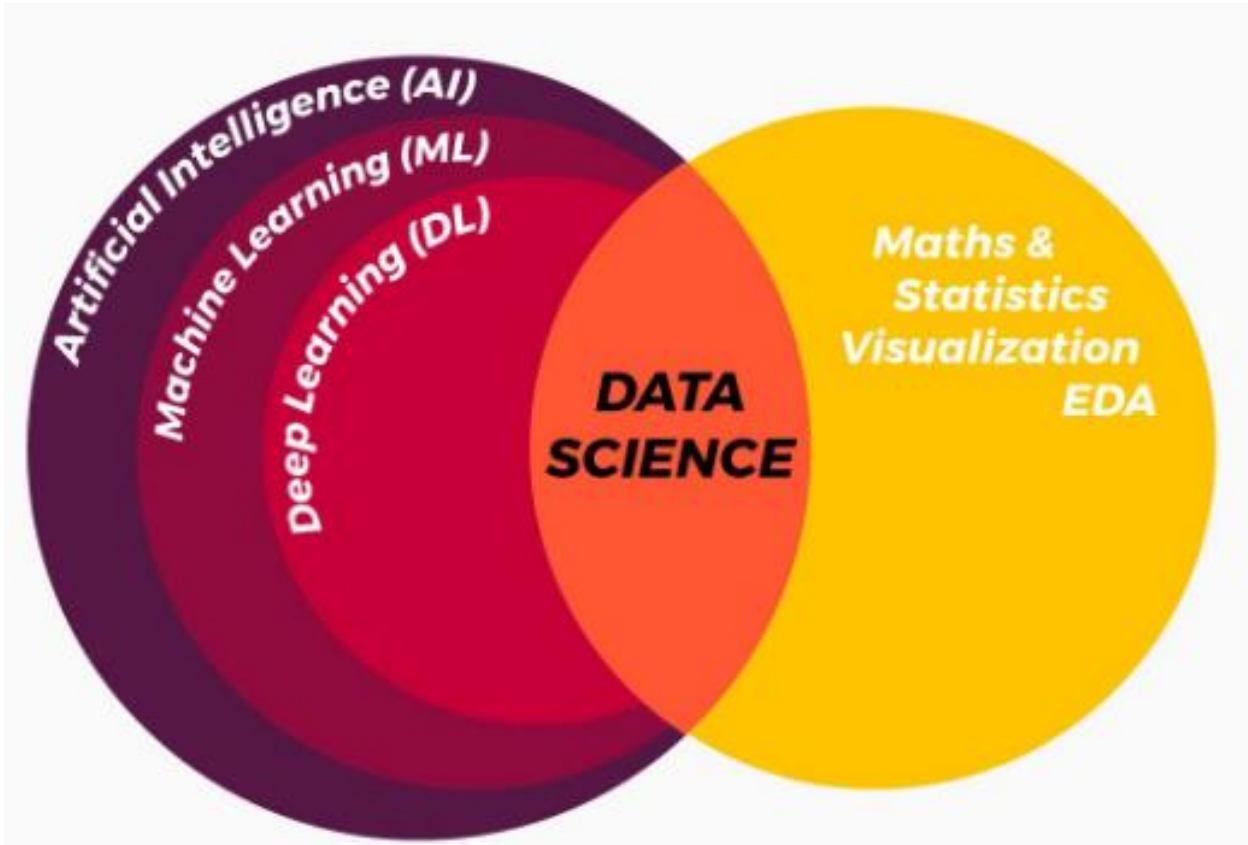


Artificial Intelligence – Machine Learning



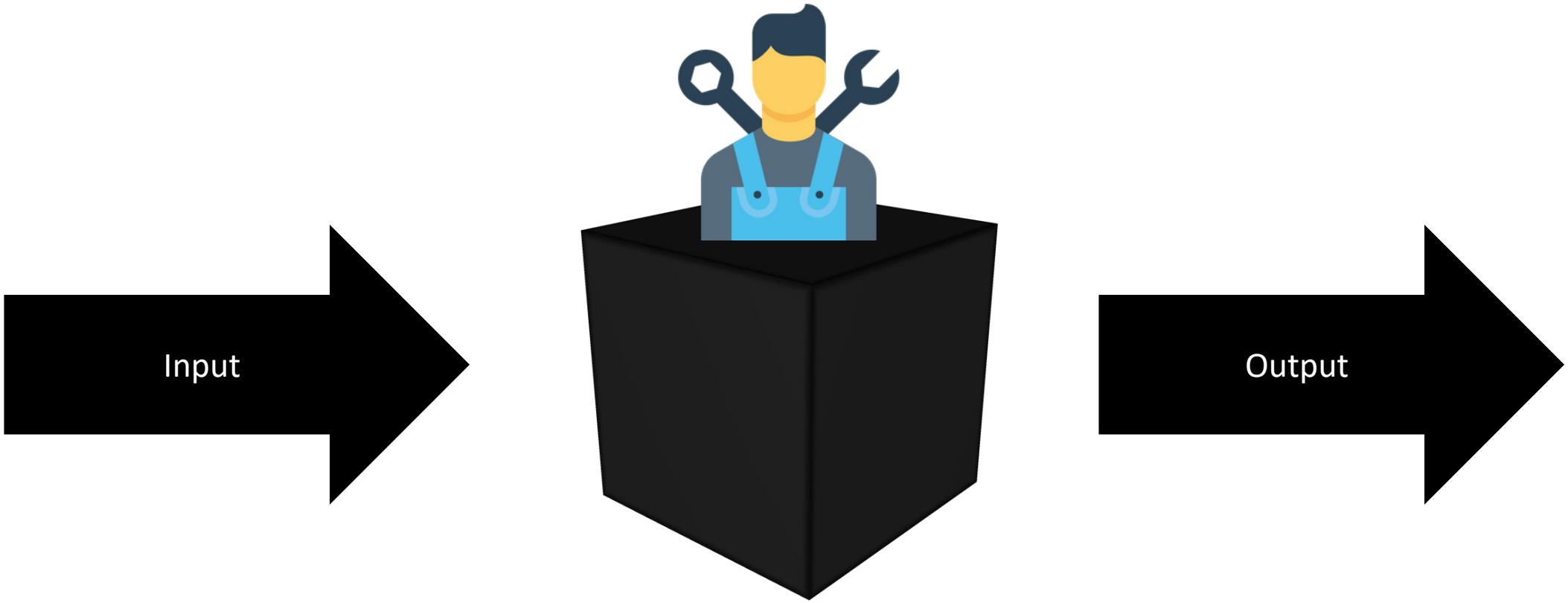


Artificial Intelligence – Machine Learning & Data Science

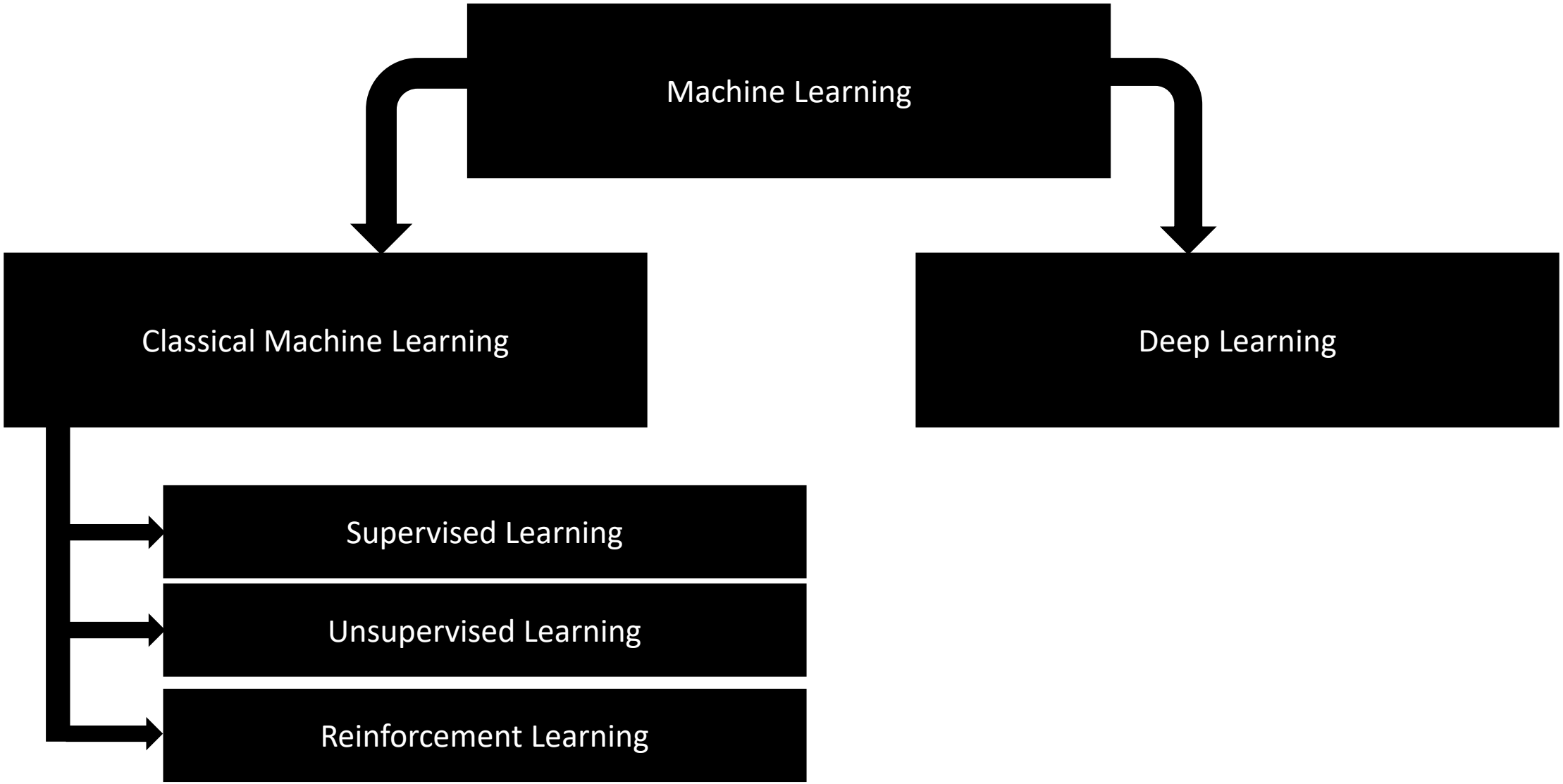


Machine Learning

Machine learning is a method of data analysis that automates analytical model building It is a branch of artificial intelligence based on the idea that systems can learn from data, identify patterns and make decisions with minimal human intervention



Types of Machine Learning Algorithms



Supervised Learning

Supervised learning is where you have input variables (X) and an output variable (Y) and you use an algorithm to learn the mapping function from the input to the output.

$$Y = f(X)$$

In supervised learning, an algorithm is trained using a training dataset and that process can be thought of as a teacher supervising the learning process.

We know the correct answers, the algorithm iteratively makes predictions on the training data and is corrected by the teacher Learning stops when the algorithm achieves an acceptable level of performance.

Accuracy of the model will be tested using testing dataset by calculating the error for the testing data.

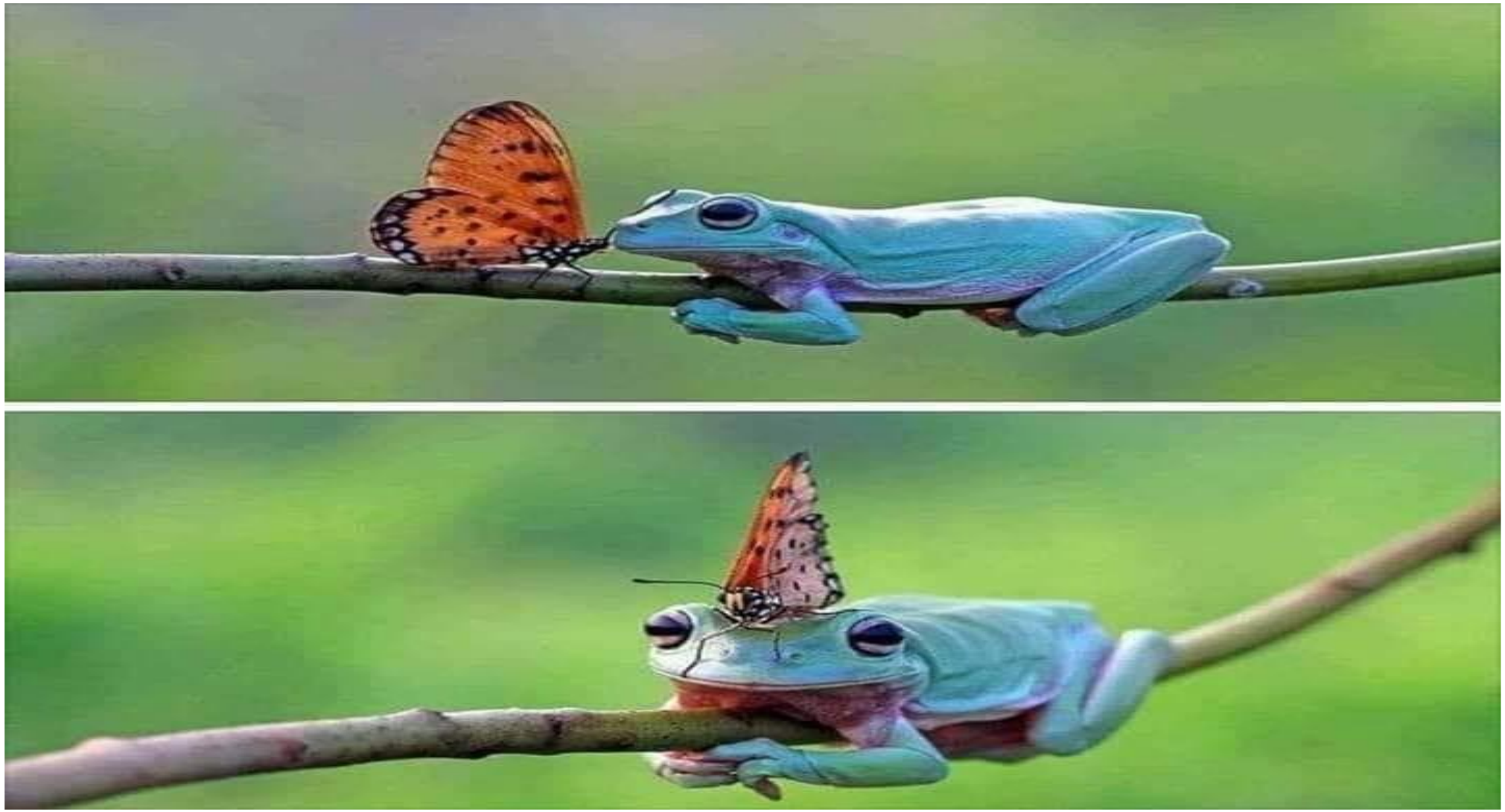
Supervised learning methods can be grouped into two categories

- **Classification** : when the output variable is a category
- **Regression** : when the output variable is a real value

Supervised Learning

Regression	Classification
Linear Regression	Logistic Regression
Regression Trees	Classification Trees
Random Forest Regression	Random Forest Classification
Extremely Randomized Trees Regression	Extremely Randomized Trees Classification
KNN Regression	KNN Classification
	Support Vector Machines
	Naïve Bayes Classification
	Bootstrapping Aggregation (Bagging)
	Boosting
	Extreme Gradient Boosting (XG Boost)

What happens next ?



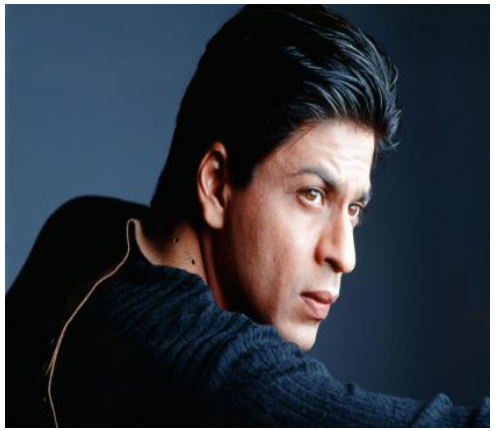
Sometimes the actual results may not be as expectations. In supervised learning these are called the errors Then the errors should be minimized

Unsupervised Learning

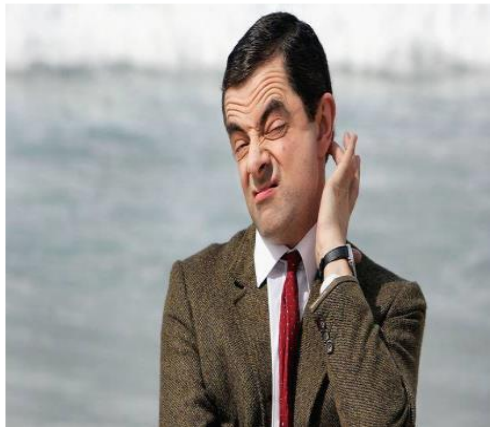
Unsupervised learning, is where you only have input data (X) and no corresponding output variables. Derive some structure or pattern from the “unlabeled data” by just looking at the relationship between the data themselves. There are few methods to check the accuracy of these models.

No teacher is in the class room. Children will play freely

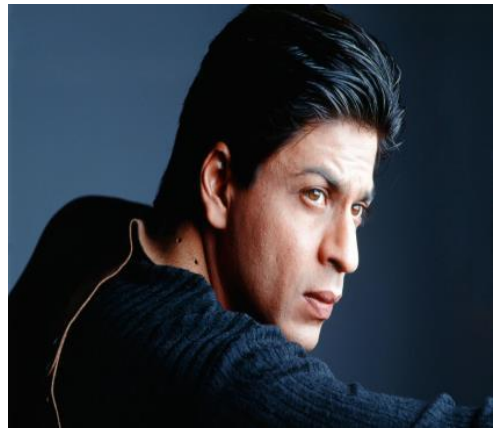




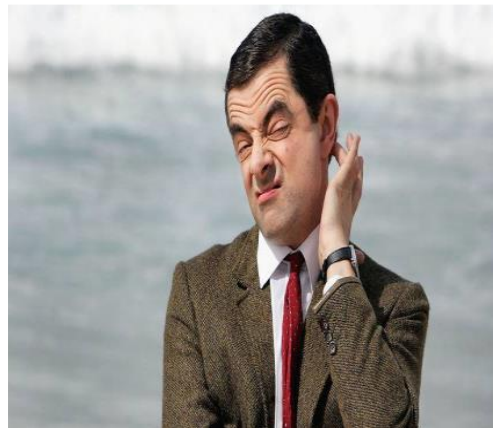




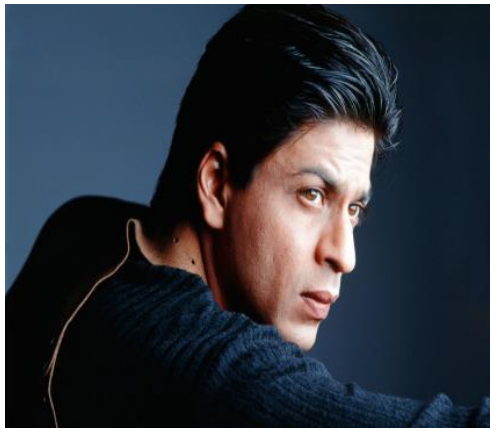




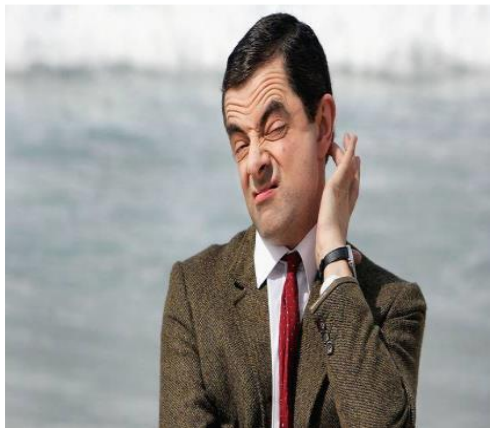




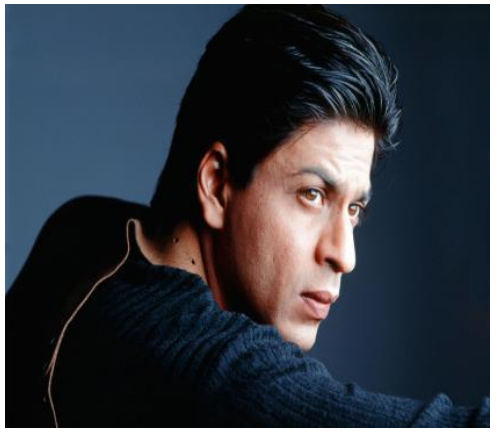




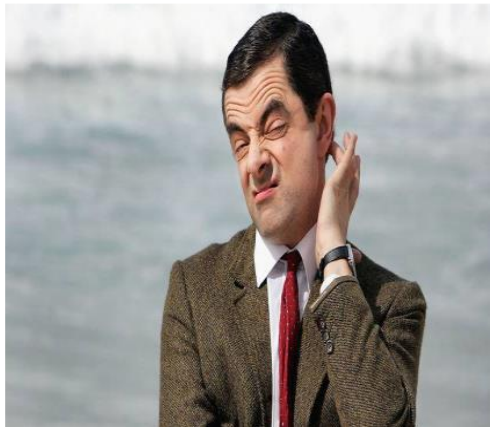


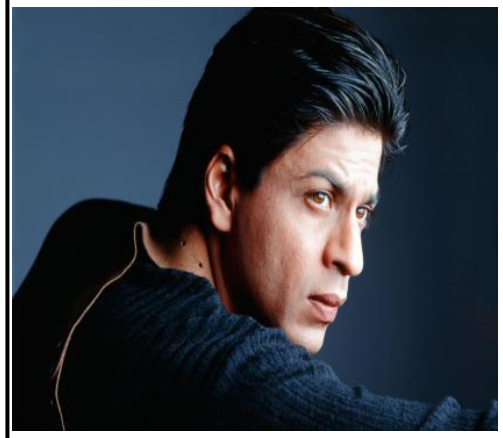






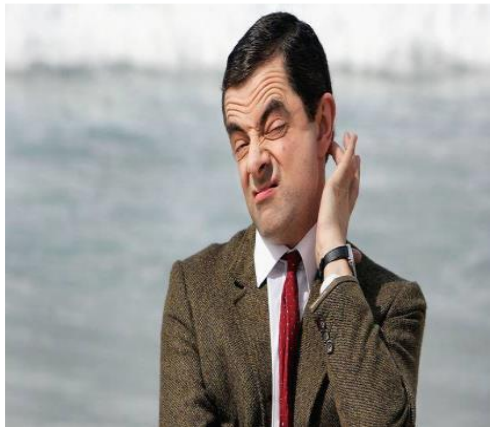


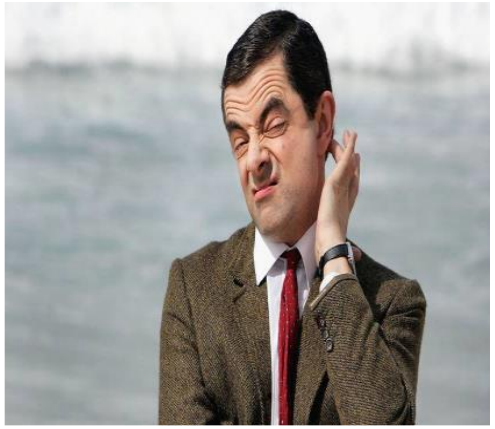


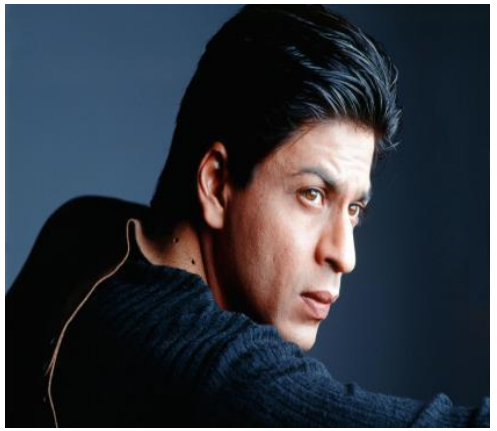











Unsupervised Learning

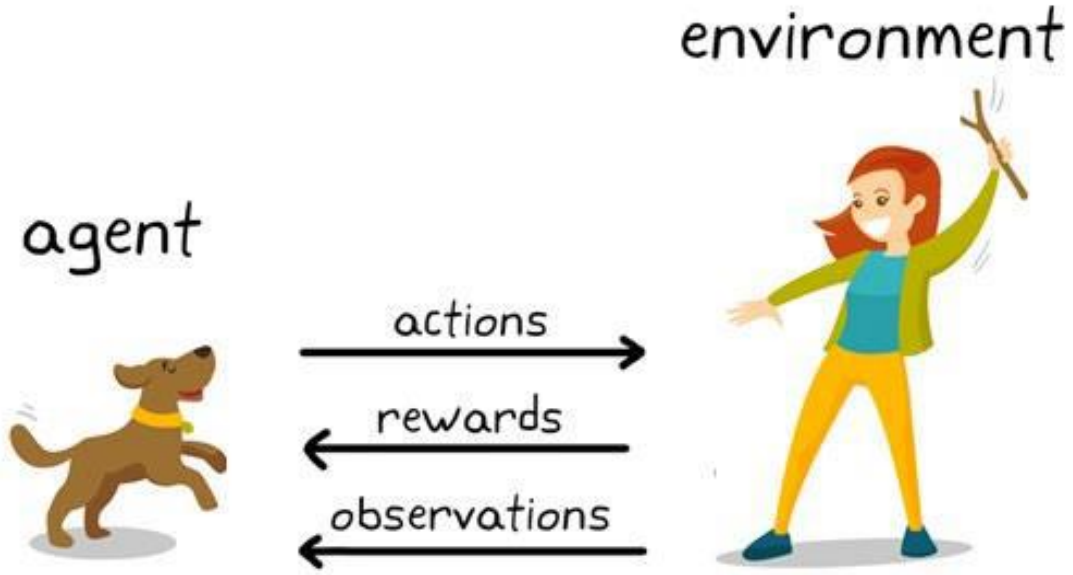
Unsupervised learning methods can be grouped into two categories.

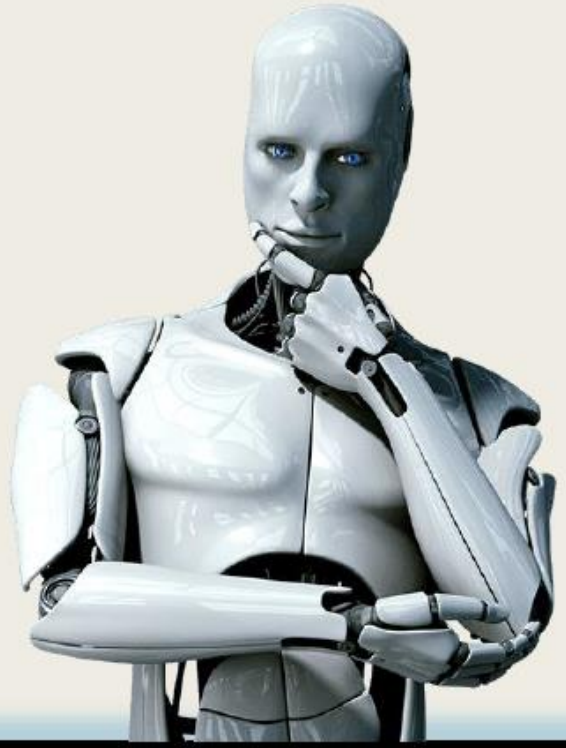
- **Clustering** : Grouping data.
  - Method by which large sets of data are grouped into clusters of smaller sets of similar data.
  - Popular Clustering algorithms are,
    - K Means algorithm
    - Hierarchical Clustering
    - Density Based Clustering
    - Learning Vector Quantization
    - Self Organizing Maps
- **Association Rule Discovery** : Identifying the associations.
  - Method of identifying associated items and identifying rules of associations.
  - The most popular method is Market Basket Analysis.

Reinforcement Learning

It is about taking suitable action to maximize reward in a particular situation. It is employed by various software and machines to find the best possible behavior or path it should take in a specific situation.

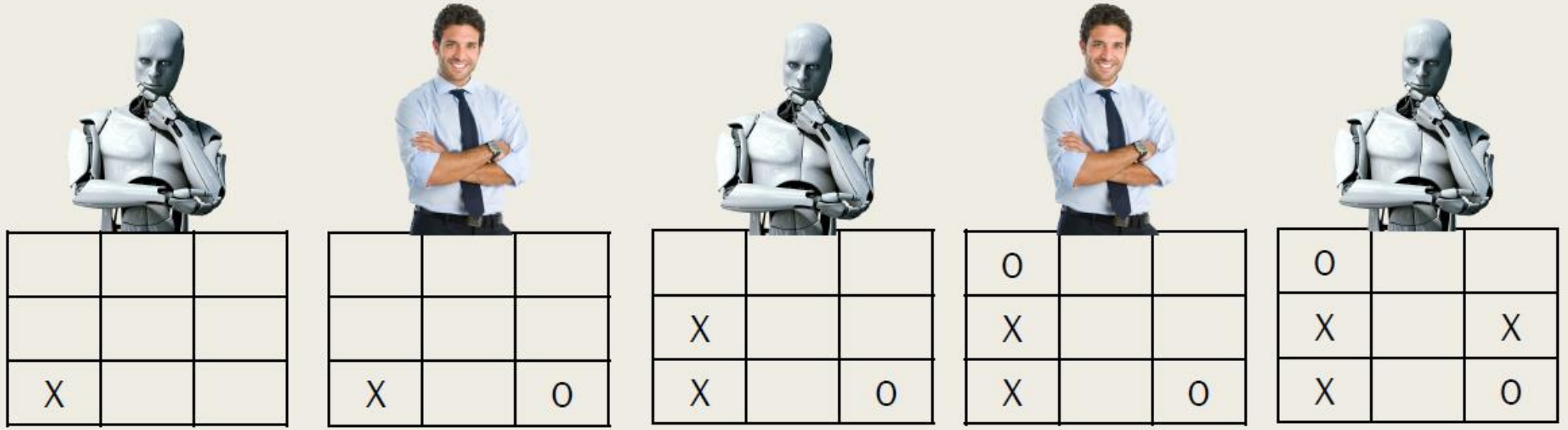
Reinforcement learning differs from the supervised learning in a way that in supervised learning the training data has the answer key with it so the model is trained with the correct answer itself whereas in reinforcement learning, there is no answer but the reinforcement agent decides what to do to perform the given task





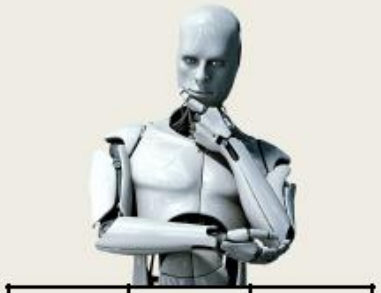
Lets Play **Tic Tac Toe**

Game 01




Learning through
mistakes

Game 02



		X

		O
		X

		O
	X	
		X

O		O
	X	
		X

O		O
	X	
X		X

O	O	O
	X	
X		X


Learning through
mistakes

Game 03

	X	

	O	
	X	

	O	
	X	
	X	

	O	
O	X	
	X	

	O	
O	X	
	X	X

O	O	
O	X	
	X	X

O	O	
O	X	
X	X	X


Never make past
mistakes

Reinforcement Learning

There are two important learning models in reinforcement learning:

- Markov Decision Process
- Q learning

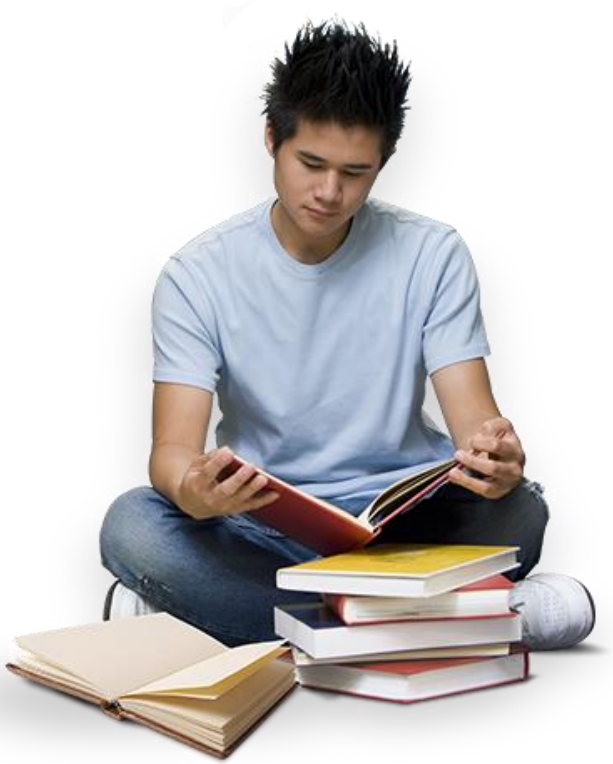
Deep Learning

Brain inspired systems which are intended to replicate the way that we humans learn. Consist of input and output layers, as well as (in most cases) a hidden layer consisting of units that transform the input into something that the output layer can use.

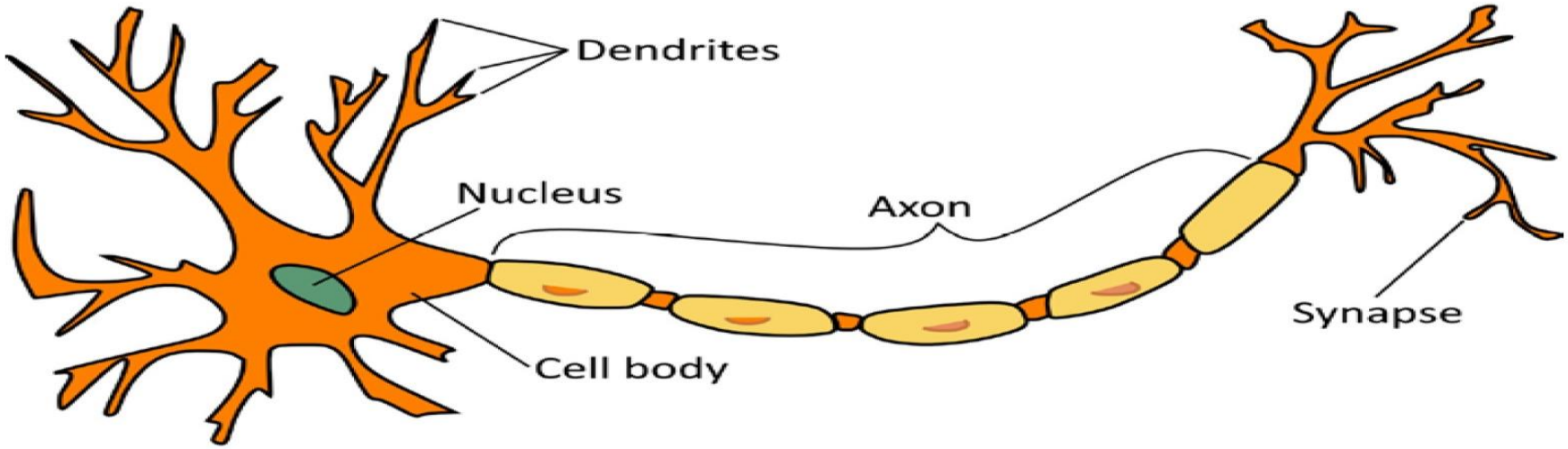
When a child is born, what does the child know? To our knowledge, the child knows only how to cry The child probably does not know its parents. When the child grows, the step by step learning process begins First, the child learns to drink milk Then the child learns to identify its parents Every time a child learns something, it is encoded into some portion of the brain. If we do not practice what we learned, we start to forget Consequently, by practice or training, we can hard code some selected things into our brains.

Deep Learning

Neuroscientists believe that learning stimulates new dendrite connections between neurons. Greater usage of the brain through learning and stimulation creates greater dendrite connectivity thus, as we learn more and more, we become more intelligent. Wisdom is not created through genetics Wisdom and knowledge are based on how we learn and how we practice what we learned.

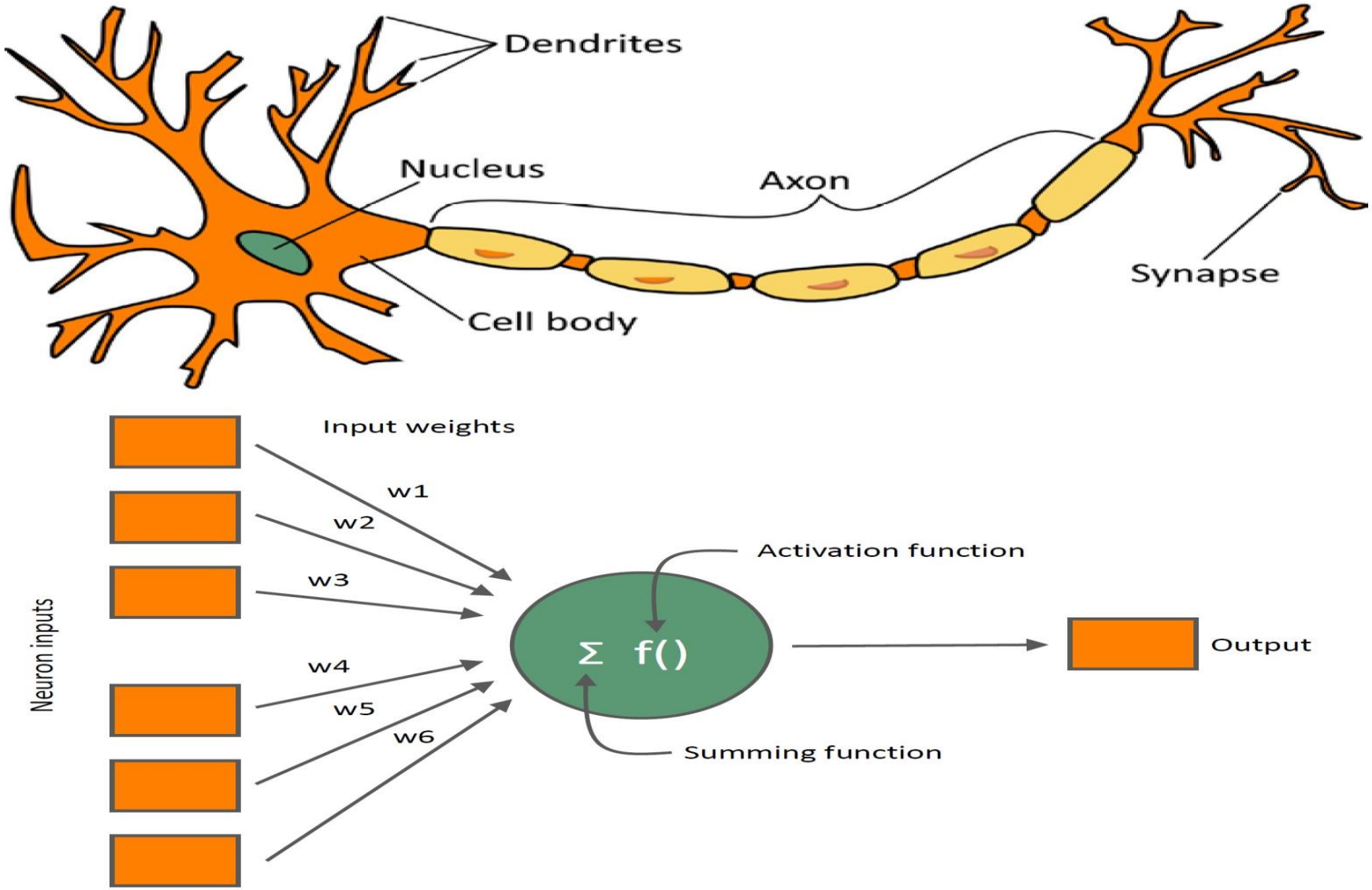


Deep Learning - Biological Neuron

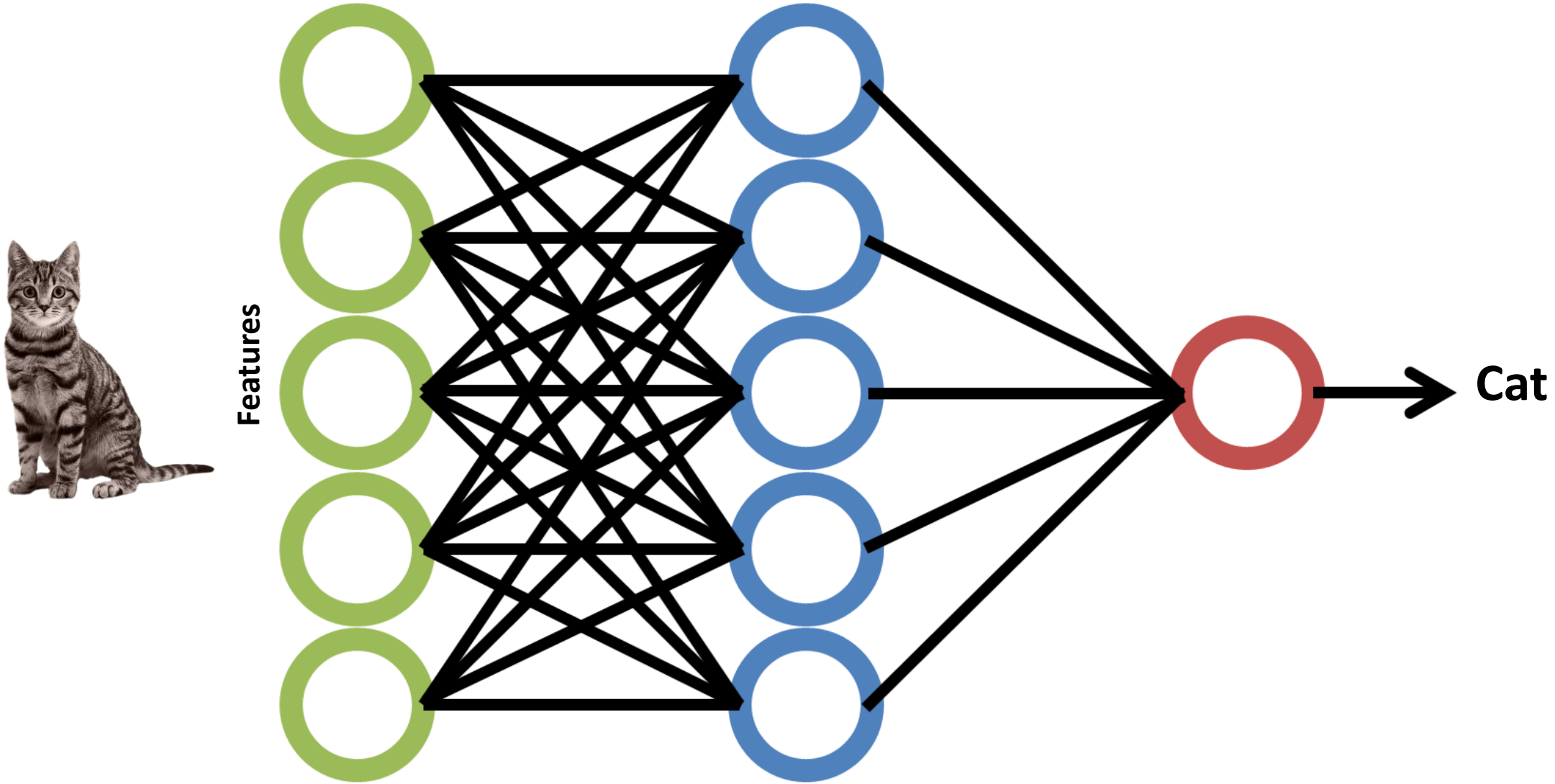


In Deep Learning, an artificial neuron is created which will follow the functions of the biological neuron.

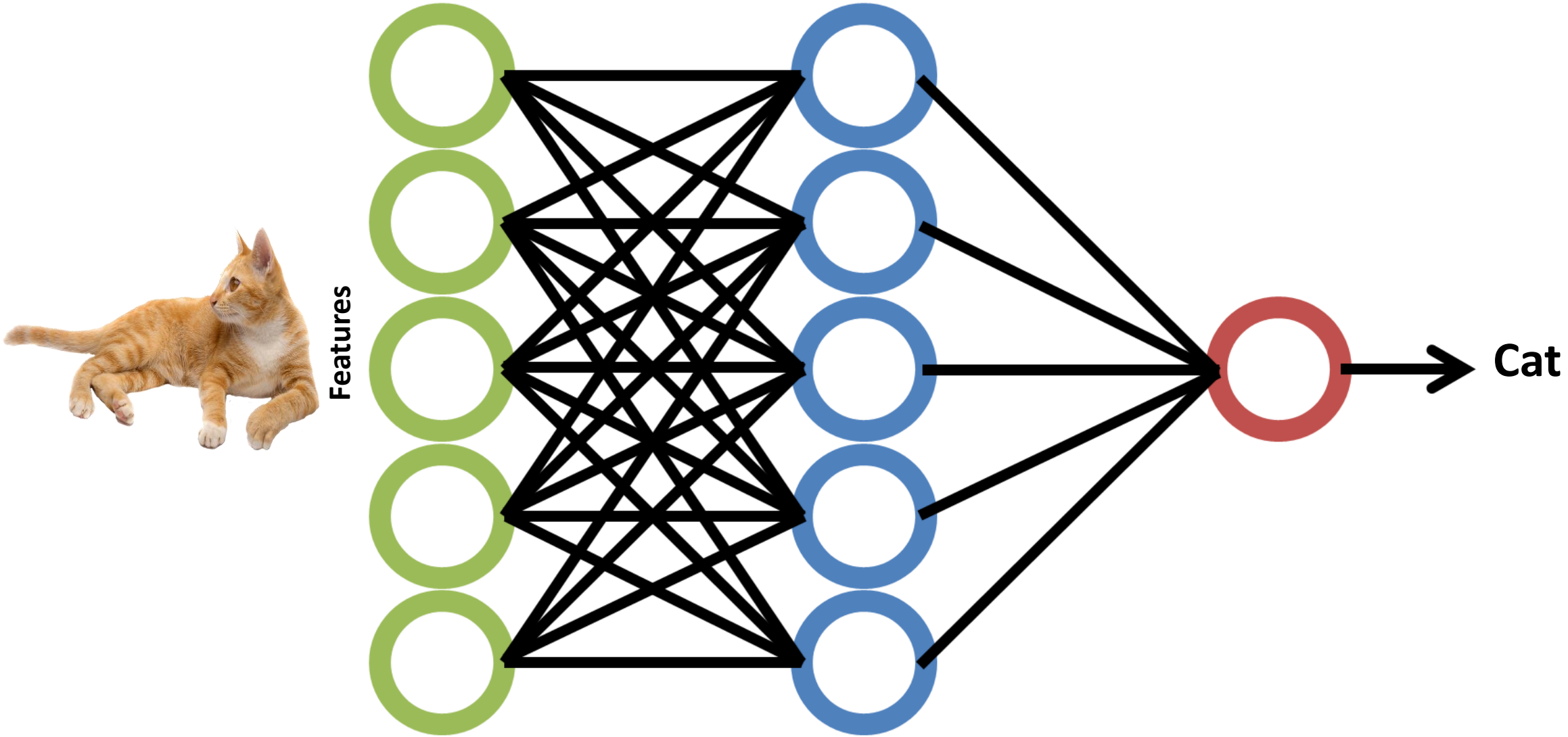
Deep Learning - Biological Neuron VS Artificial Neuron



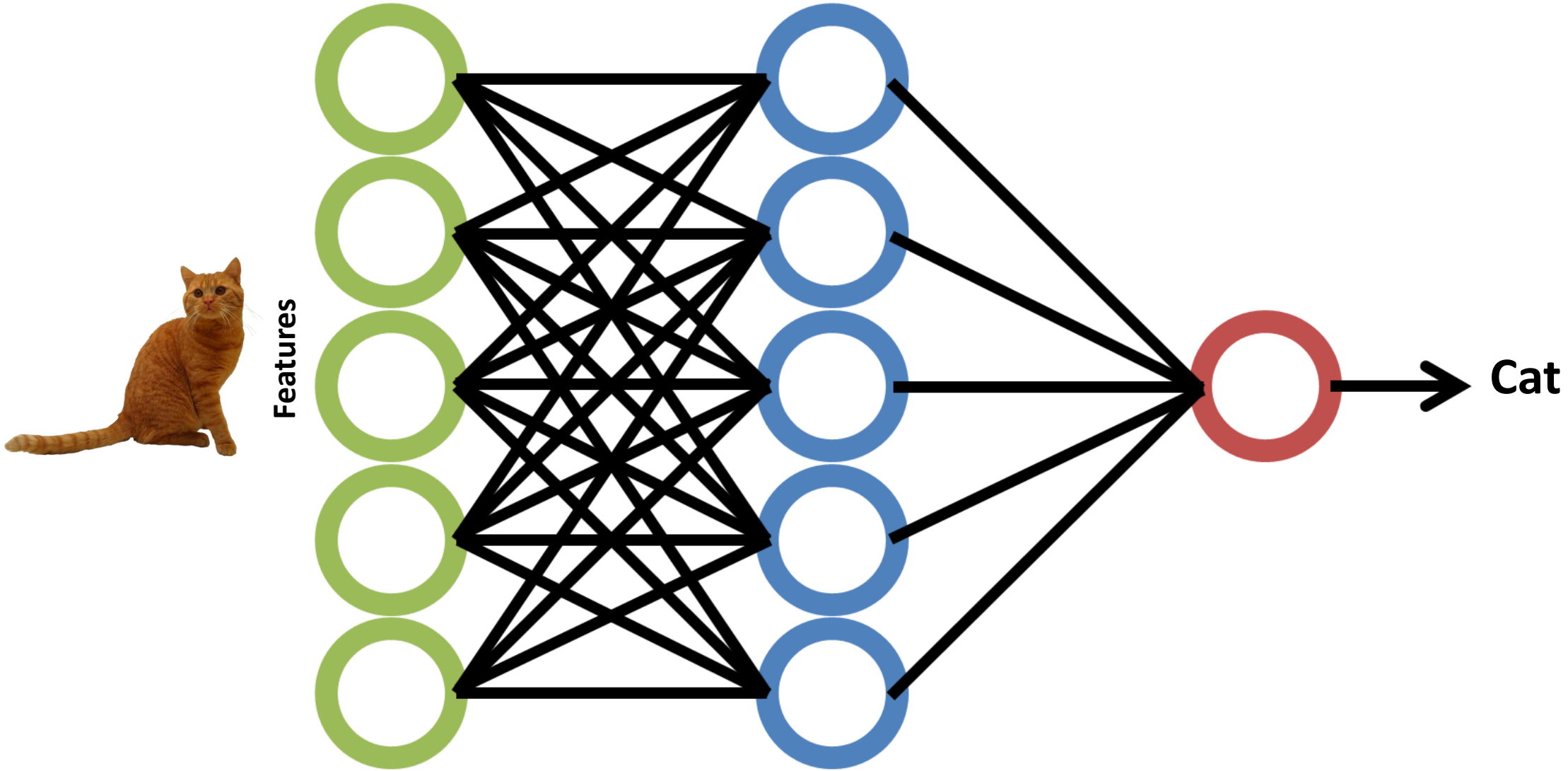
Deep Learning – Learning Process



Deep Learning – Learning Process



Deep Learning – Learning Process



Deep Learning – Learning Process

Trained System

What should be returned as the output?

Deep Learning – Learning Process

Trained System

Cat

Applications of Machine Learning – Banking Industry

Credit Scorecard Model

- Credit Scorecards are basically used to assess the credit worthiness of customers.
- These are heavily used in the industry for taking decisions on granting credit, monitoring portfolio and calculating expected loss.
- Logistic Regression, LDA, Decision Trees, Random Forest, Bagging and Boosting can be used for classifying the good or bad customers

Stock Price Forecasting Model

- These models are used to predict price of a stock or an index for a given time period in future
- Getting stock price of any of the publicly listed companies.
- Data is known as univariate time series data.
- ARIMA class of models or Exponential Smoothing models can be used for predicting.

Applications of Machine Learning – Banking Industry

Segmentation Modeling

- Segmentation Modeling is used to carter differently to different segments of customers.
- The historical data on customer attributes data on financial products services are used to build the segmentation models.
- Mostly the Decision Trees and Clustering techniques are used.

Revenue Forecasting

- Regression analysis can be used for forecasting the revenue with the factors affecting to the revenue for a set of periods of equal interval (Quarterly, Half Year, Annually).

Pricing Financial Products

- Models are built to price financial products such as mortgages, auto loans, credit card and transactions.
- Most of the companies are building models for price forwards, futures, options as well as swaps.

Applications of Machine Learning – Banking Industry

Prepayment Modeling

- The models are build to know if a customer prepays, when is he likely to prepay in the life time of the loan

Fraud Model

- These models are fitted to know if a particular transaction is a fraudulent transaction.
- Mostly the Logistic Regression and Decision Trees are used.

Applications of Machine Learning – Insurance Industry

Consumer Targeting

- Targeting intendant customers in a better way.
- Searching on the customer data.
- Recommending the best products to the customers.

Risk Assessment and Pricing

- Models are built to quantify the risk of the customer and setting appropriate premium for the risk.

Customer Relationship Management

- Building models for identifying the behavior of the customers and to build the relationships.
- Fast track quick settlements (Prioritizing).
- Automatic approvals.

Applications of Machine Learning – Marketing Industry

Customer Segmentation

- Clustering gives marketers a new ability to discover customer segments.
- Most advanced analysis requires a very clear understanding of the relationship of data.
- For most novice data enabled marketers, this is understandably a challenge without an infusion of renewed insights.
- Clustering techniques make that journey easier to arrive at, reaching the right segments to plan and deploy relevant offerings.

Customer Retention

- Considering the rise of customer success, it's invariable that algorithms or models for retention would have a great deal of investment.
- In this case, Logistic Regression takes independent variables and determines the likelihood of a prospect/customer to churn.
- Customer churn is a challenge because there are more unpredictable variables including customer reps engagement community response factors.

Applications of Machine Learning – Marketing Industry

Demand Forecasting

- Demand forecasting which is an area of predictive analytics that looks to provide future estimates of products or services to be consumed or used.
- It goes beyond educated guesses and looks at historical sales data or current data from test markets.
- Analytic techniques such as Time Series Analysis are mostly used in these cases.

Applications of Machine Learning –Retail Industry

Loyalty Analysis

- Finding factors responsible for loyalty.
- More about Statistical Inference.

Campaign Analysis (Marketing Research)

- Analysis the data collected from marketing campaign.

Market Basket Analysis

- Which products are purchased simultaneously.

Segmentation Analysis

Personalized Recommendation

Supervised Learning – Linear Regression

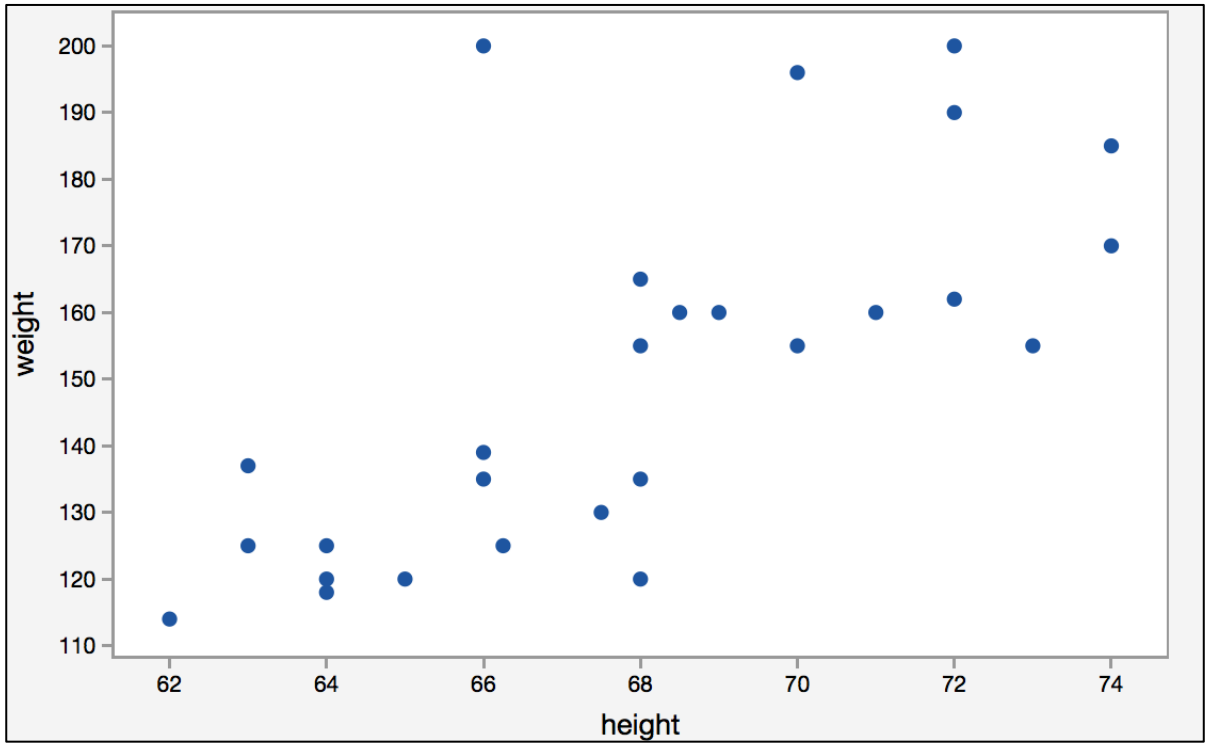
Linear regression is perhaps one of the most well known and well understood algorithms in statistics and machine learning.

Linear regression is a linear model, e.g. a model that assumes a linear relationship between the input variables (x) and the single output variable (y). More specifically, that y can be calculated from a linear combination of the input variables (x).

When there is a single input variable (x), the method is referred to as **Simple Linear Regression**. When there are multiple input variables, literature from statistics often refers to the method as **Multiple Linear Regression**.

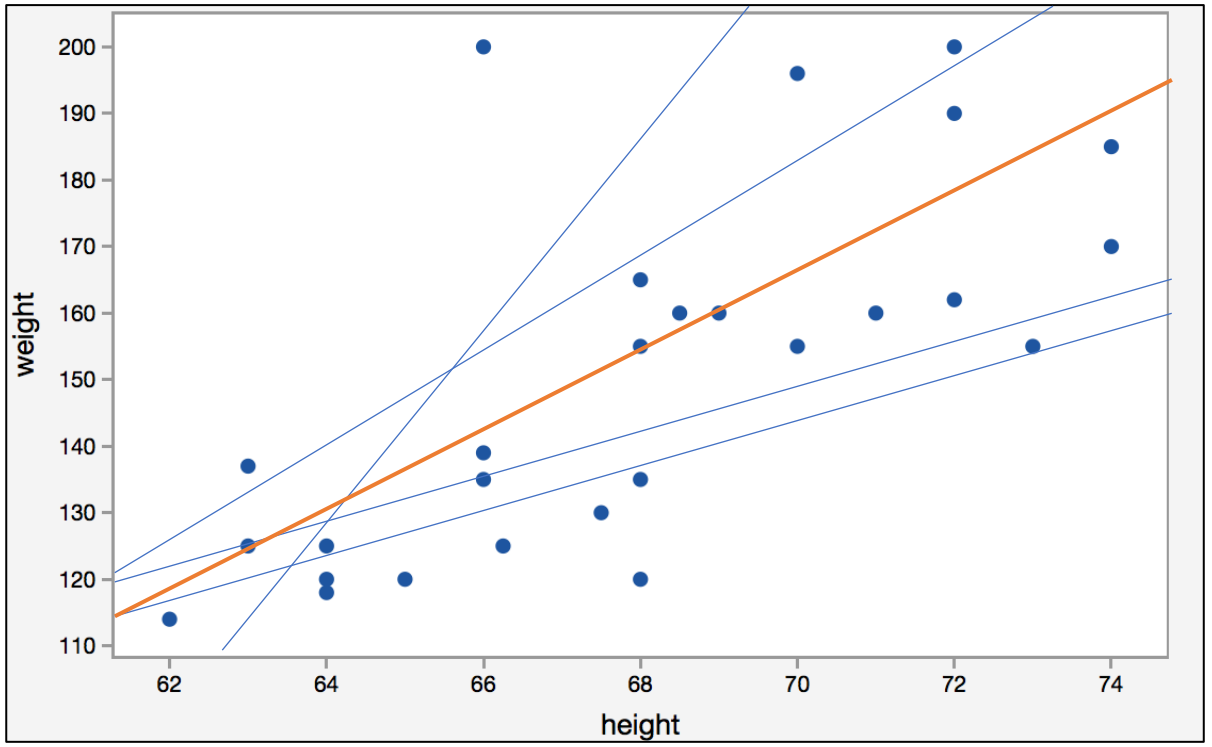
Different techniques can be used to prepare or train the linear regression equation from data, the most common of which is called **Ordinary Least Squares**. It is common to therefore refer to a model prepared this way as Ordinary Least Squares Linear Regression or just Least Squares Regression.

Supervised Learning – Simple Linear Regression



Simple linear regression is useful for finding relationship between two variables. One is predictor or independent variable and other is response or dependent variable which is a quantitative variable. For example, relationship between height and weight.

Supervised Learning – Simple Linear Regression



The core idea is to obtain a line that best fits the data. The best fit line is the one for which total prediction error (all data points) are as small as possible. Error is the distance between the point to the regression line.

Supervised Learning – Simple Linear Regression

The equation for this model is as follows,

$$y = \beta_0 + \beta_1 x + \varepsilon$$

The values β_0 and β_1 must be chosen so that they minimize the error. If sum of squared error is taken as a metric to evaluate the model, then goal to obtain a line that best reduces the error.

$$Sum\ of\ Squares\ of\ Error\ (SSE) = \sum_{i=1}^{n} (Actual\ Output - Predicted\ Output)^2 = \sum_{i=1}^{n} (y_i - \hat{y}_i)^2$$

This is called as Residual Sum of Squares (RSS) as well. By minimizing this SSE, we can obtain following parameter estimations. Significance of these parameters can also be measured using **Hypothesis Testing**.

$$\hat{\beta}_1 = \frac{\sum_{i=1}^{n} (x_i - \bar{x})\,(y_i - \bar{y})}{\sum_{i=1}^{n} (x_i - \bar{x})^2}$$

$$\hat{\beta}_0 = \bar{y} - \hat{\beta}_1 \bar{x}$$

Then the estimated model is,

$$\hat{y} = \hat{\beta}_0 + \hat{\beta}_1 x$$

Supervised Learning – Multiple Linear Regression

The equation for this model is as follows,

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_3 + \cdots + \beta_k x_k + \varepsilon$$

Consider here we have k variables. Then the parameter vector $\bar{\beta}$ can be obtained through,

$$\bar{\beta} = (X^T X)^{-1} X^T Y$$

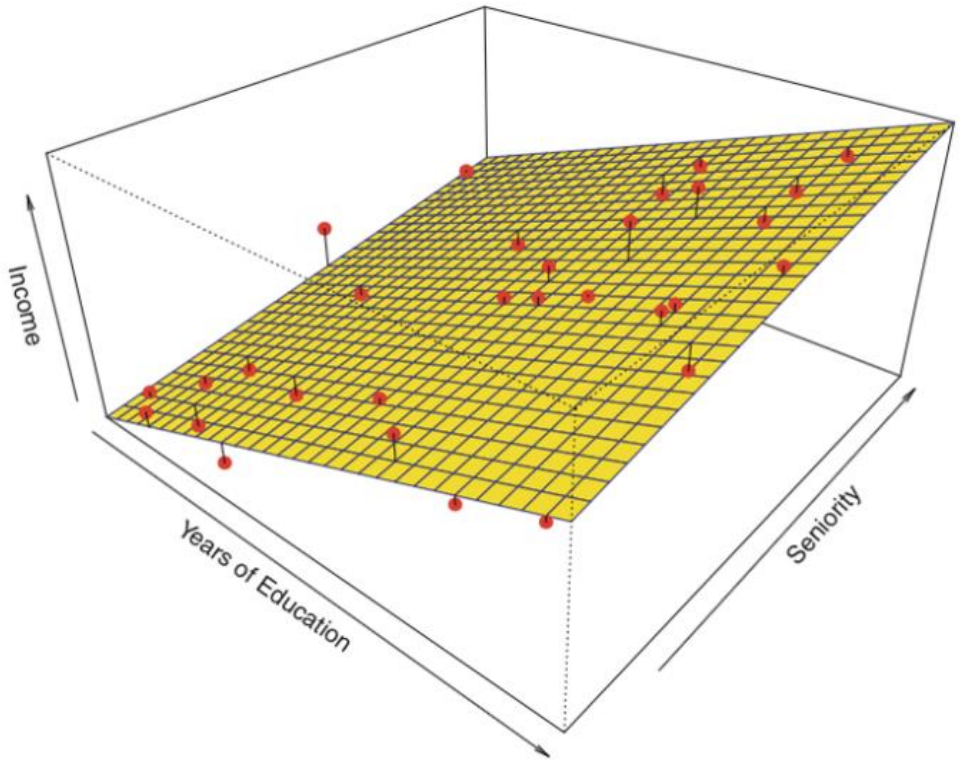
As mentioned above the significance of these variables can also be checked. P- Value is calculated using Hypothesis Testing. If P-Value is less than 0.05 (Significance Level) the parameter is significant. Further the significance of the model can also be checked.

Here,

$$\mathbf{X} = \begin{bmatrix} 1 & X_{11} & X_{12} & \cdots & X_{1k} \\ 1 & X_{21} & X_{22} & \cdots & X_{2k} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ 1 & X_{n1} & X_{n2} & \cdots & X_{nk} \end{bmatrix}$$

Supervised Learning – Multiple Linear Regression

How this model is visualized. Consider a 2 variables with one response example. Here the **Income** is the response variable and **Seniority** and the **Years Of Education** are the independent variables.



Higher dimensions cannot be visualized easily. There are several advanced techniques for visualize them.

Supervised Learning – R Squared Value (Coefficient of Determination)

$$Total\ Sum\ of\ Squares\ (TSS) = \sum_{i=1}^{n}(y_i - \bar{y})^2$$

$$Sum\ of\ Squares\ of\ Error\ (SSE) = \sum_{i=1}^{n}(y_i - \hat{y}_i)^2$$

$$Coefficient\ of\ Determination = R^2 = \frac{TSS - SSE}{TSS}$$

This R Squared Value is explaining the fraction of variation explained by the estimated model. In simple words how much of the data captured by this model.

For the Simple Linear Regression case $Coefficient\ of\ Determination = R^2 = (Correlation)^2$

Supervised Learning –Linear Regression Assumptions

There are mainly four assumptions associated with a linear regression model:

- **Linearity**: The relationship between X and the mean of Y is linear.
    - Scatter plots & partial regression models
- **Multicollinearity**: Correlation among independent variables.
    - VIF
- **Homoscedasticity**: The variance of residual is the same for any value of X.
    - Levene's test
- **Independence**: Observations are independent of each other.
    - Residuals VS fitted values
- **Normality**: For any fixed value of X, Y is normally distributed.
    - Q-Q plot & Shapiro–Wilk test

Statistics puts an emphasis on model inference, while Machine Learning puts an emphasis on accurate predictions. So, these assumptions have no much impact in Machine Learning.

Supervised Learning –Dummy Variables

To represent categorical variables in a linear regression model, we use dummy variables. Consider following example.

Ex:- Gender

Gender	Dummy Variable (G)
Male	1
Female	0

Ex- Temperature (High, Medium, Low)

Temperature	Dummy Variable (T1)	Dummy Variable (T2)
High	1	0
Medium	0	1
Low	0	0

Ex- Colour (Red, Green, Yellow, Blue)

Colour	Dummy Variable (C1)	Dummy Variable (C2)	Dummy Variable (C3)
Red	1	0	0
Green	0	1	0
Yellow	0	0	1
Blue	0	0	0

$$\hat{y} = \hat{\beta}_0 + \hat{\beta}_1 x_1 + \hat{\beta}_2 c_1 + \hat{\beta}_3 c_2 + \hat{\beta}_4 c_3$$

Supervised Learning – Model Evaluation

There are mainly two techniques for evaluating a model in regression.

- Validation Set Approach
- Cross Validation

Using these approaches we can measure some metrics for evaluating a model. The most popular metric among these metrics is,

$$Mean\ Squared\ Error = \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{n}$$

Supervised Learning – Validation Set Approach

Split the dataset into two sets,

- Training dataset (Generally 80 of the data But it can be changed)
- Testing dataset (Rest of the data)

Train the model using the training dataset and then check the accuracy of the model using the testing dataset. MSE of a regression model will be calculated and the model will be evaluated.

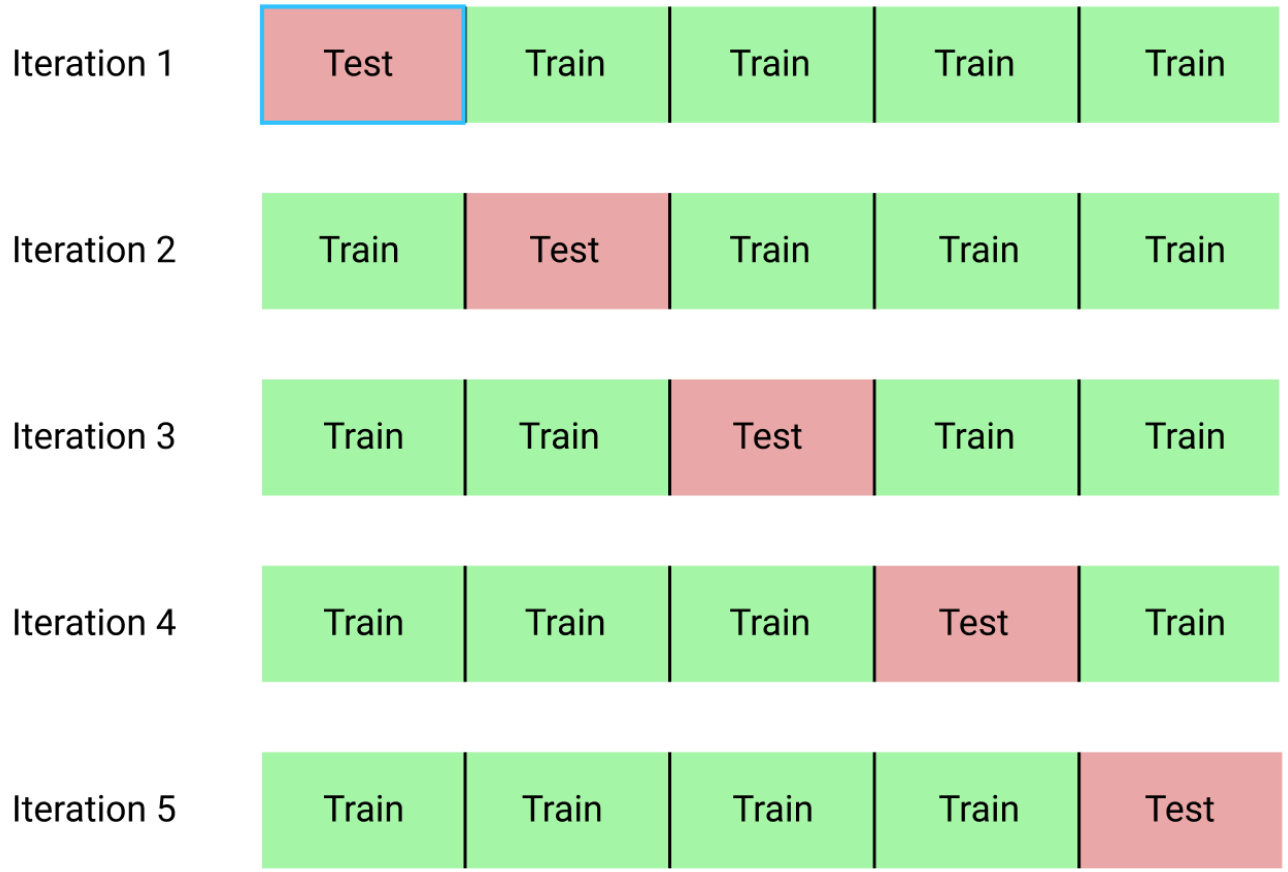
Supervised Learning – Overfitting & Underfitting

Overfitting refers to a model that models the training data too well. This means that the noise or random fluctuations in the training data is picked up and learned as concepts by the model. The problem is that these concepts do not apply to new data and negatively impact the models ability to generalize.

Underfitting refers to a model that can neither model the training data nor generalize to new data. An underfit machine learning model is not a suitable model and will be obvious as it will have poor performance on the training data.

Supervised Learning – Cross Validation Approach

Cross-validation, sometimes called rotation estimation or out-of-sample testing, is any of various similar model validation techniques for assessing how the results of a statistical analysis will generalize to an independent data set.



Supervised Learning –Feature Selection

Feature selection is the process of selecting the most important variables for fitting the model among a set of independent variables. There are 5 techniques for selecting the best set of features for a model.

- Best Subset Selection
- Forward Selection
- Backward Selection
- Recursive Feature Elimination

Supervised Learning –Feature Selection

Best Subset Selection

- Compute least squared fit for all possible subsets and then choose the best among them based on some criterion that balances training error & model size.
- 2^P possible models are there if the number of variables is P. When the P increases, number of models is increased highly.

Forward Selection

- Start with null model.
- Fit P simple linear regression models and get the lowest RSS model.
- Increase that model with rest of the variables (P-1) with the lowest RSS.
- Construct until some stopping rule is satisfied.

Supervised Learning –Feature Selection

Backward Selection

- Start with the model with all P variables.
- Remove one variable and select the most significant model (P-1).
- Continue until a stopping rule is reached

Model Selection

Selecting the best model among the models produced by forward or backward or best subset selection.

- Validation Set or Cross Validation Approach (Direct Approach)
- Mallow's C_p (Best model will give, C_p ≈Number of parameters)
- Akaike Information Criteria (AIC) (Lowest value will give the best)
- Bayesian Information Criteria (BIC) (Lowest value will give the best)
- Adjusted R Squared (Largest value will give the best)

Indirect Approaches

Supervised Learning –Feature Selection

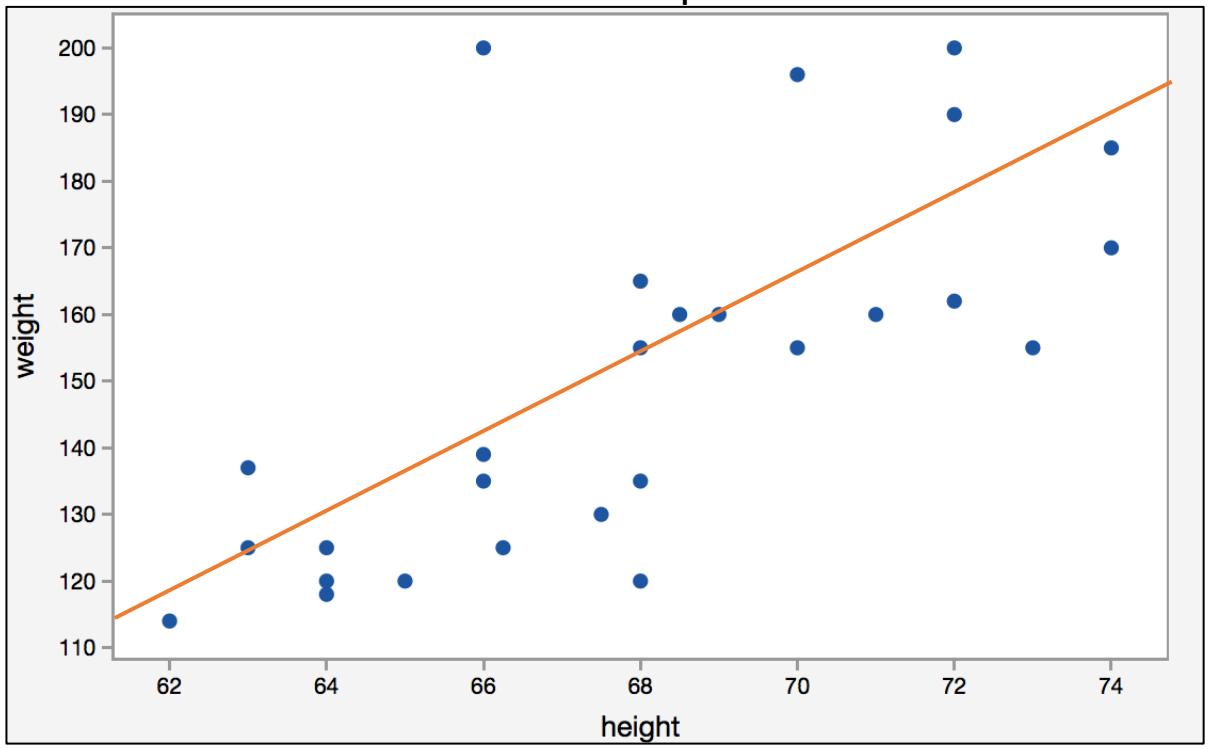
Recursive Feature Elimination

- In Python, above discussed techniques are not available.
- Instead of these techniques, Python provides a better way which is called the Recursive Feature Elimination (RFE).
- The goal of recursive feature elimination (RFE) is to select features by recursively considering smaller and smaller sets of features.
- First, the estimator is trained on the initial set of features and the importance of each feature is obtained either through any specific attribute or callable.
- Then, the least important features are pruned from current set of features.
- That procedure is recursively repeated on the pruned set until the desired number of features to select is eventually reached.

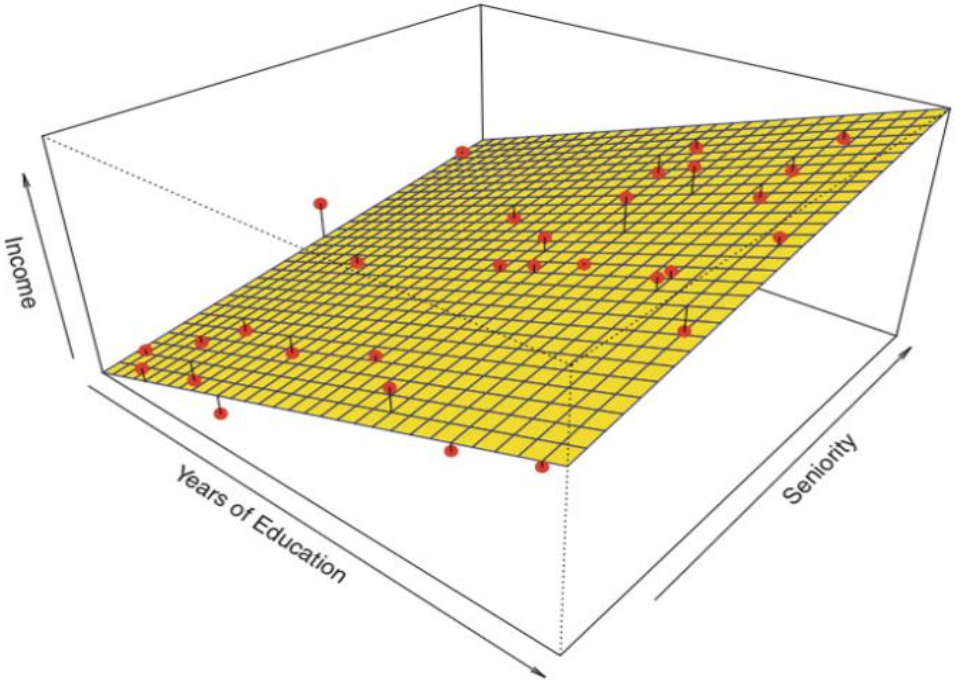
Dimensionality Reduction

Dimensions are the directions of variables. Number of variables is equal to the number of dimensions of a regression model. We have two popular techniques; **Principal Component Analysis** and **Factor Analysis**.

Consider all these variables as independent variables with some response variable.



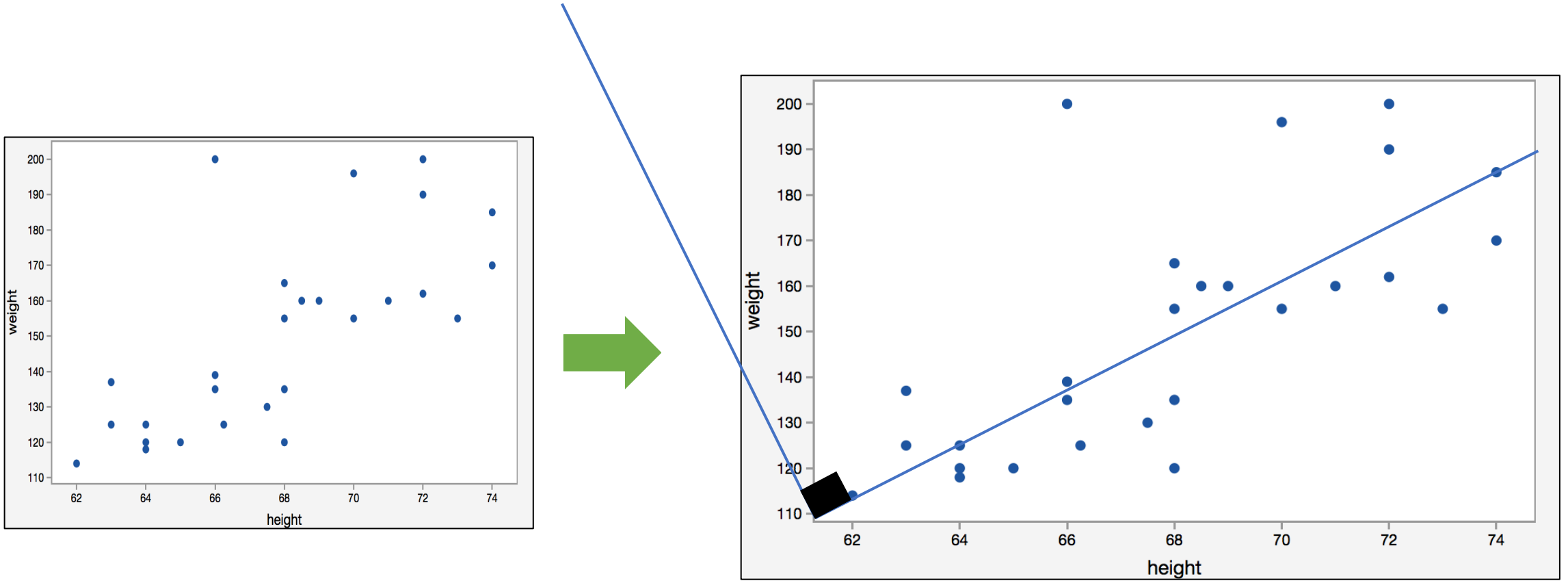
Two dimensions



Three dimensions

Dimensionality Reduction - PCA

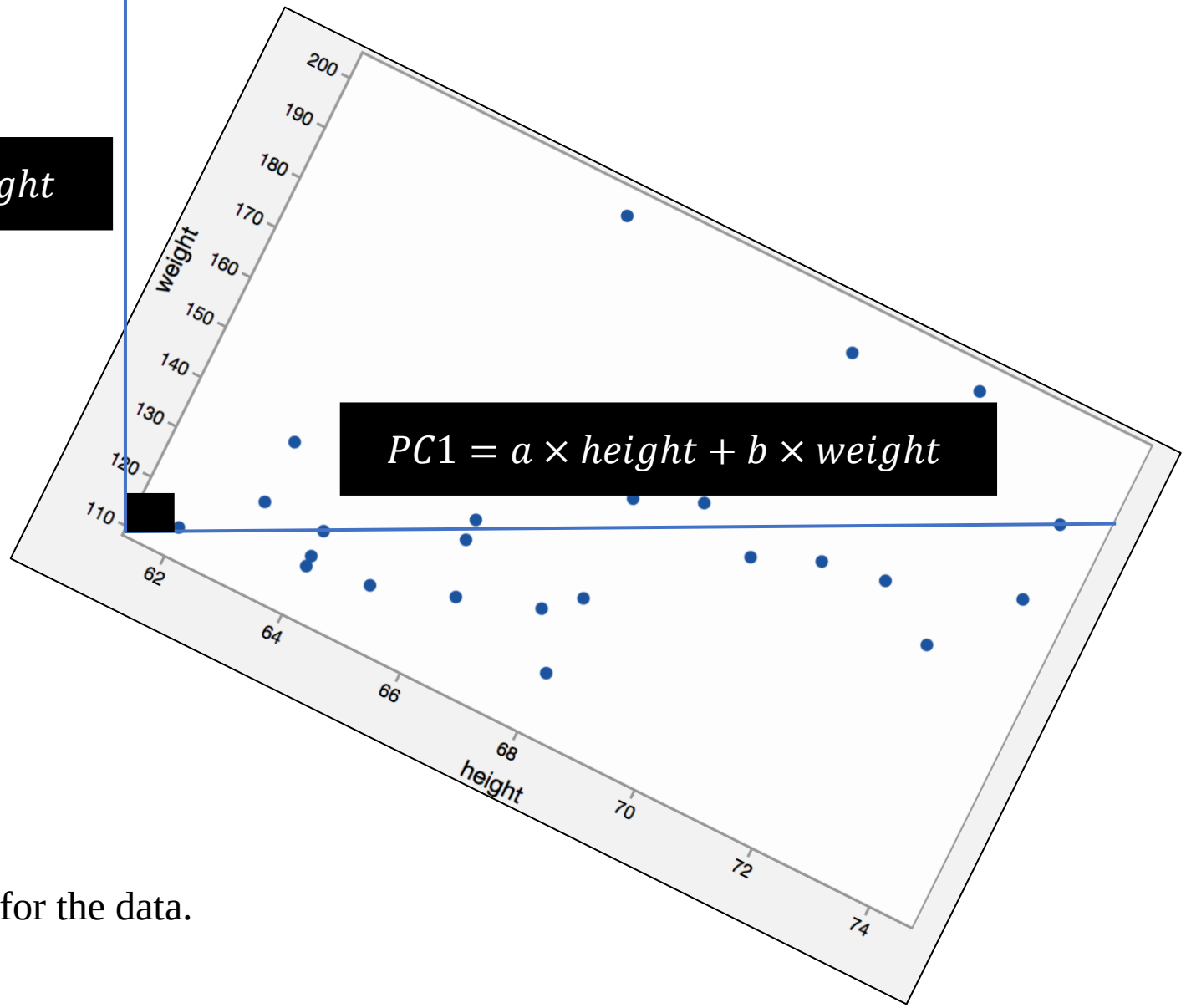
We can reduce these dimensions using the correlations among the variables.



Find the axis along the highest correlation and find the second axis which is uncorrelated to the first axis. These axes can be written as linear combinations of original variables. These are called **Principal Components**.

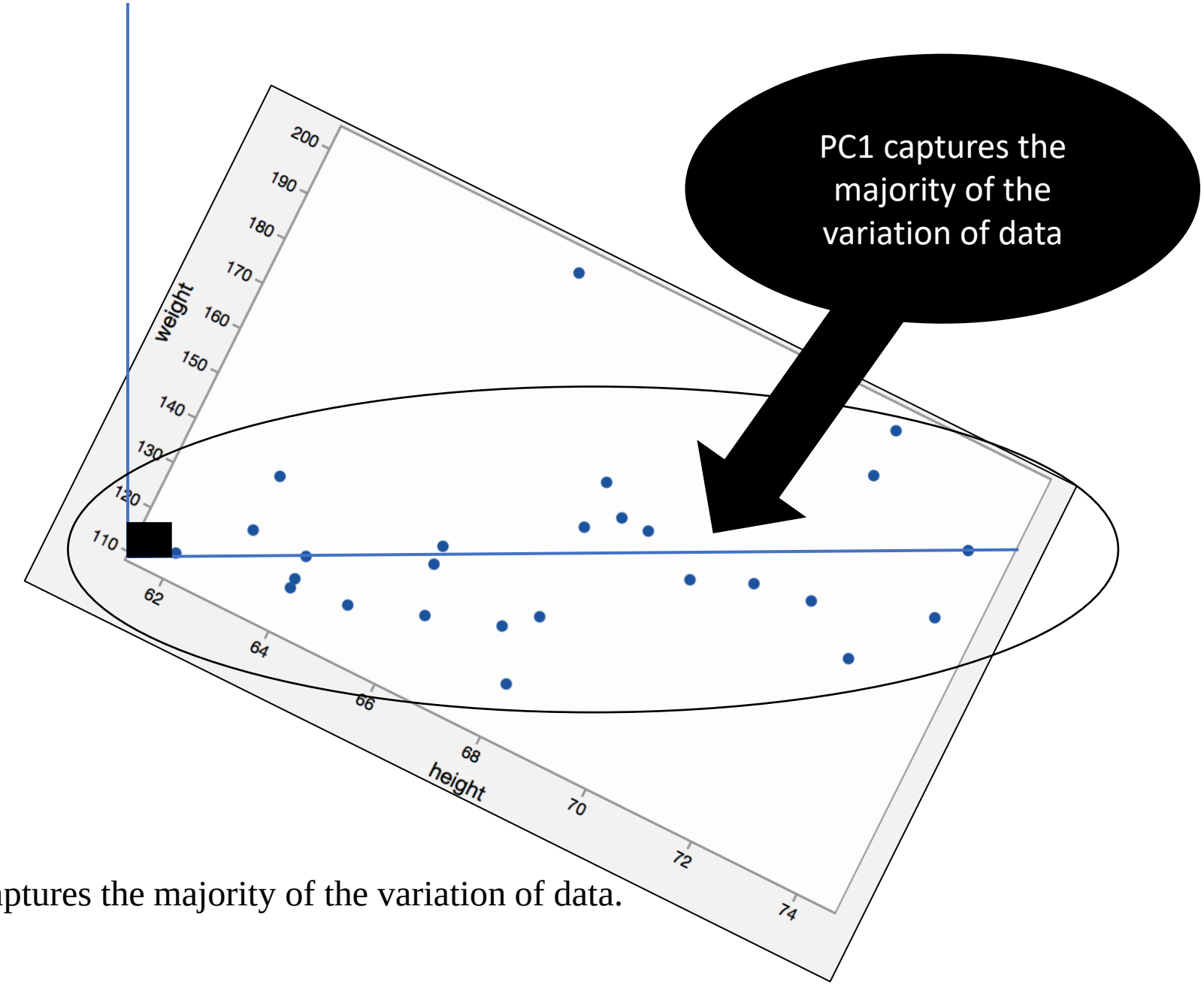
Dimensionality Reduction - PCA

$$PC2 = c \times height + d \times weight$$



These are the new axes for the data.

Dimensionality Reduction - PCA



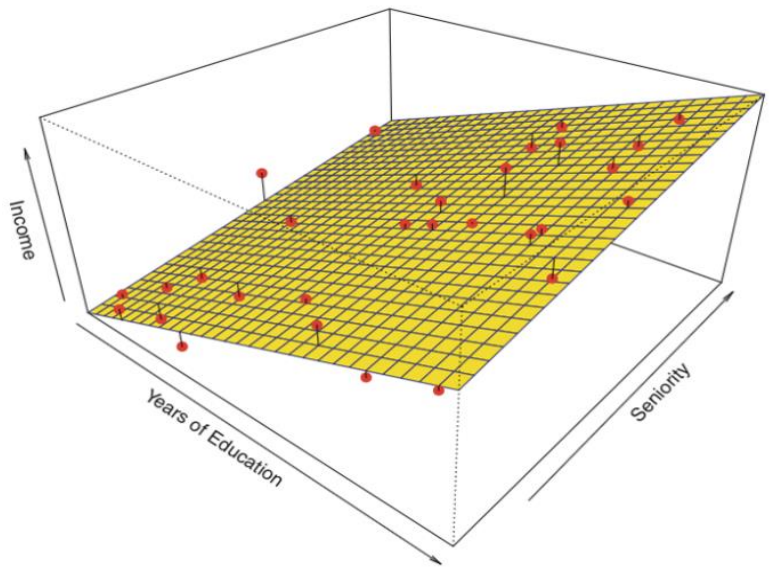
Select the PC which captures the majority of the variation of data.

Dimensionality Reduction - PCA

Now the dimensions of the data have been reduced to 1. Since PC1 captures the majority (Above 90%) variation of the data. So now we can go with PC1 newly created variable instead of height and weight two variables.

$$\hat{y} = \hat{\beta}_0 + \hat{\beta}_1 PC1$$

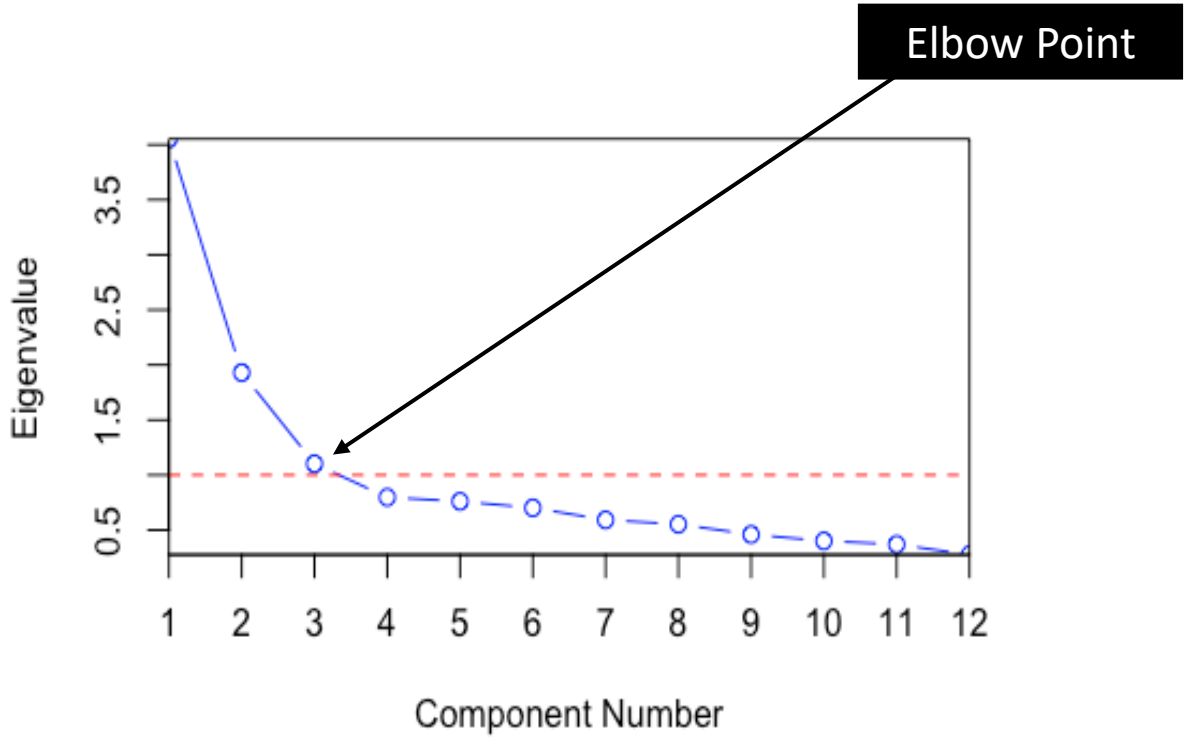
What about other cases where the dimensions above 2?



Using the same approach, we can find PC1, PC2, PC3 principal components. Then we can select the PC s which are capturing the majority of the variation (Above 90%)

Dimensionality Reduction - PCA

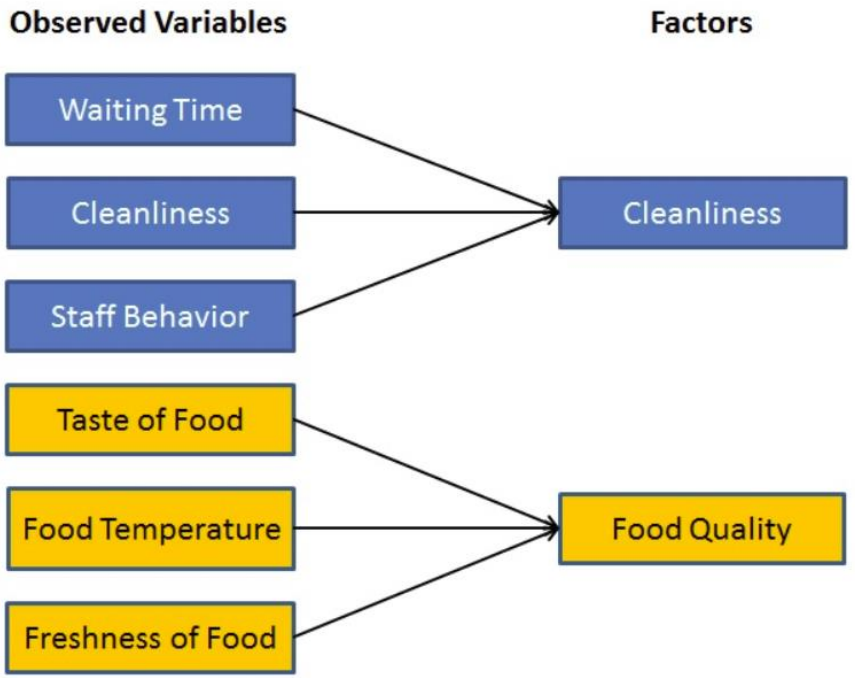
Actually these variances captured by each component are the eigenvalues of the Variance – Co Variance matrix of variables. Using them, we can select PC s which capture the majority of the variation using the scree plot. Consider the following scree plot as an example.



We can select the PC s above the Elbow Point. We can select the PC s using the cumulative variance as well.

Dimensionality Reduction - FA

Factor analysis is a linear statistical model. It is used to explain the variance among the observed variable and condense a set of the observed variable into the unobserved variable called factors. Observed variables are modeled as a linear combination of factors and error terms (Source). Factor or latent variable is associated with multiple observed variables, who have common patterns of responses. Each factor explains a particular amount of variance in the observed variables. It helps in data interpretations by reducing the number of variables.



Dimensionality Reduction - FA

The primary objective of factor analysis is to reduce the number of observed variables and find unobservable variables. These unobserved variables help the market researcher to conclude the survey. This conversion of the observed variables to unobserved variables can be achieved in two steps:

Factor Extraction: In this step, the number of factors and approach for extraction selected using variance partitioning methods such as principal components analysis and common factor analysis.

Factor Rotation: In this step, rotation tries to convert factors into uncorrelated factors — the main goal of this step to improve the overall interpretability. There are lots of rotation methods that are available such as: Varimax rotation method, Quartimax rotation method, and Promax rotation method.

Dimensionality Reduction - FA

Factor Loadings

- The factor loading is a matrix which shows the relationship of each variable to the underlying factor. It shows the correlation coefficient for observed variable and factor. It shows the variance explained by the observed variables.

Eigenvalues

- Eigenvalues represent variance explained each factor from the total variance. This can be used for selecting the number of factors as in PCA.

Communality

- Commonalities are the sum of the squared loadings for each variable. It represents the common variance.

Dimensionality Reduction - FA

Factor analysis is a way to take a mass of data and shrinking it to a smaller data set that is more manageable and more understandable. It's a way to find hidden patterns, show how those patterns overlap and show what characteristics are seen in multiple patterns. It is also used to create a set of variables for similar items in the set (these sets of variables are called dimensions).

We can find factor scores using a technique called Factor Analysis and we can create a matrix as the right. Consider the example.

This shows that 7 factors capture all 32 variables as sub groups. As PCA, here also factor variables can be created. These are called the Latent variables.

Ex:- $F_5 = 0.658\,X_6 + 0.833\,X_7 + 0.750\,X_{10}$

Variable	1	2	3	4	5	6	7
X1	0.200	-0.081	0.064	0.104	0.162	0.850	0.178
X2	0.168	0.161	0.099	-0.079	0.258	0.826	0.136
X3	0.674	0.077	0.116	0.126	0.211	0.358	0.083
X4	0.505	0.099	-0.238	0.458	0.218	0.294	0.090
X5	0.573	0.259	-0.103	-0.004	0.223	0.328	0.357
X6	0.372	-0.096	-0.064	0.275	0.658	0.240	0.302
X7	0.281	0.139	0.131	0.199	0.833	0.103	0.194
X8	0.214	-0.024	0.198	0.020	0.200	0.090	0.793
X9	0.612	0.194	0.196	-0.117	0.392	-0.019	0.277
X10	0.233	0.238	0.224	0.127	0.750	0.164	0.002
X11	0.410	0.049	0.202	0.524	0.358	0.127	0.159
X12	0.284	0.067	0.607	0.081	0.170	0.219	0.491
X13	0.696	-0.064	0.403	0.222	0.101	-0.094	0.086
X14	0.714	0.208	0.122	0.192	0.135	0.261	-0.008
X15	0.251	0.207	0.137	0.743	0.272	0.133	0.001
X16	0.176	0.321	0.038	0.866	0.029	0.048	-0.064
X17	-0.084	0.515	0.301	0.567	0.222	0.099	0.144
X18	0.094	0.783	0.133	0.155	0.065	0.154	0.072
X19	-0.025	0.618	0.163	0.209	0.372	0.415	0.026
X20	0.007	0.555	0.306	0.241	0.060	-0.029	0.477
X21	0.017	0.216	0.244	0.232	-0.064	0.706	-0.157
X22	0.215	0.633	0.030	0.232	-0.253	0.467	-0.224
X23	0.339	0.603	0.348	0.073	0.447	0.029	-0.116
X24	0.517	0.625	0.196	0.128	0.292	0.015	-0.217
X25	0.505	0.443	0.239	0.301	0.349	-0.045	-0.111
X26	0.452	0.598	0.110	0.283	0.001	-0.205	0.327
X27	0.123	0.024	0.700	-0.273	0.040	0.033	0.191
X28	0.056	0.197	0.848	0.099	0.155	0.173	0.118
X29	0.119	0.347	0.798	0.222	0.103	0.097	0.033
X30	-0.006	0.267	0.595	0.425	0.037	0.163	0.207
X32	0.204	0.061	0.629	0.189	0.072	-0.008	-0.390

Regularization

One of the major aspects of training your machine learning model is avoiding overfitting. The model will have a low accuracy if it is overfitting. This happens because your model is trying too hard to capture the noise in your training dataset.

Regularization is a form of regression, that constrains/ regularizes or shrinks the coefficient estimates towards zero. In other words, this technique discourages learning a more complex or flexible model, so as to avoid the risk of overfitting.

Consider the following model,

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_3 + \cdots + \beta_p x_p$$

We know that,

$$RSS = \sum_{i=1}^{n} (y_i - \hat{y}_i)^2$$

Regularization

This can be written as,

$$\mathrm{RSS} = \sum_{i=1}^{n} \left(y_i - \beta_0 - \sum_{j=1}^{p} \beta_j x_{ij} \right)^2$$

Now the model will be trained using training data. If there is noise in the training data, then the estimated coefficients won't generalize well to the future data. This is where regularization comes in and shrinks or regularizes these learned estimates towards zero.

There are two techniques for regularization,

- Ridge Regression
- LASSO Regression

Regularization – Ridge Regression (L2 Norm)

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 the model.

$$\sum_{i=1}^{n}\left(y_i-\beta_0-\sum_{j=1}^{p}\beta_j x_{ij}\right)^2+\lambda\sum_{j=1}^{p}\beta_j^2=\mathrm{RSS}+\lambda\sum_{j=1}^{p}\beta_j^2$$

Here the data should be standardized before performing the Ridge Regression.

$$\tilde{x}_{ij}=\frac{x_{ij}}{\sqrt{\frac{1}{n}\sum_{i=1}^{n}(x_{ij}-\overline{x}_j)^2}}.$$

Regularization – LASSO Regression (L1 Norm)

Lasso is another variation, in which the above function is minimized. Its clear that this variation differs from ridge regression only in penalizing the high coefficients.

$$\sum_{i=1}^{n}\left(y_i - \beta_0 - \sum_{j=1}^{p}\beta_j x_{ij}\right)^2 + \lambda\sum_{j=1}^{p}|\beta_j| = \mathrm{RSS} + \lambda\sum_{j=1}^{p}|\beta_j|.$$

Regularization

If we summarize the things we have discussed about the regularization

Ridge Regression : Minimize RSS Such that $\sum_{i=1}^{n} \beta_i^2 \leq S$

LASSO Regression : Minimize RSS Such that $\sum_{i=1}^{n} |\beta_i| \leq S$

Consider the two independent variables case (X_1, X_2).

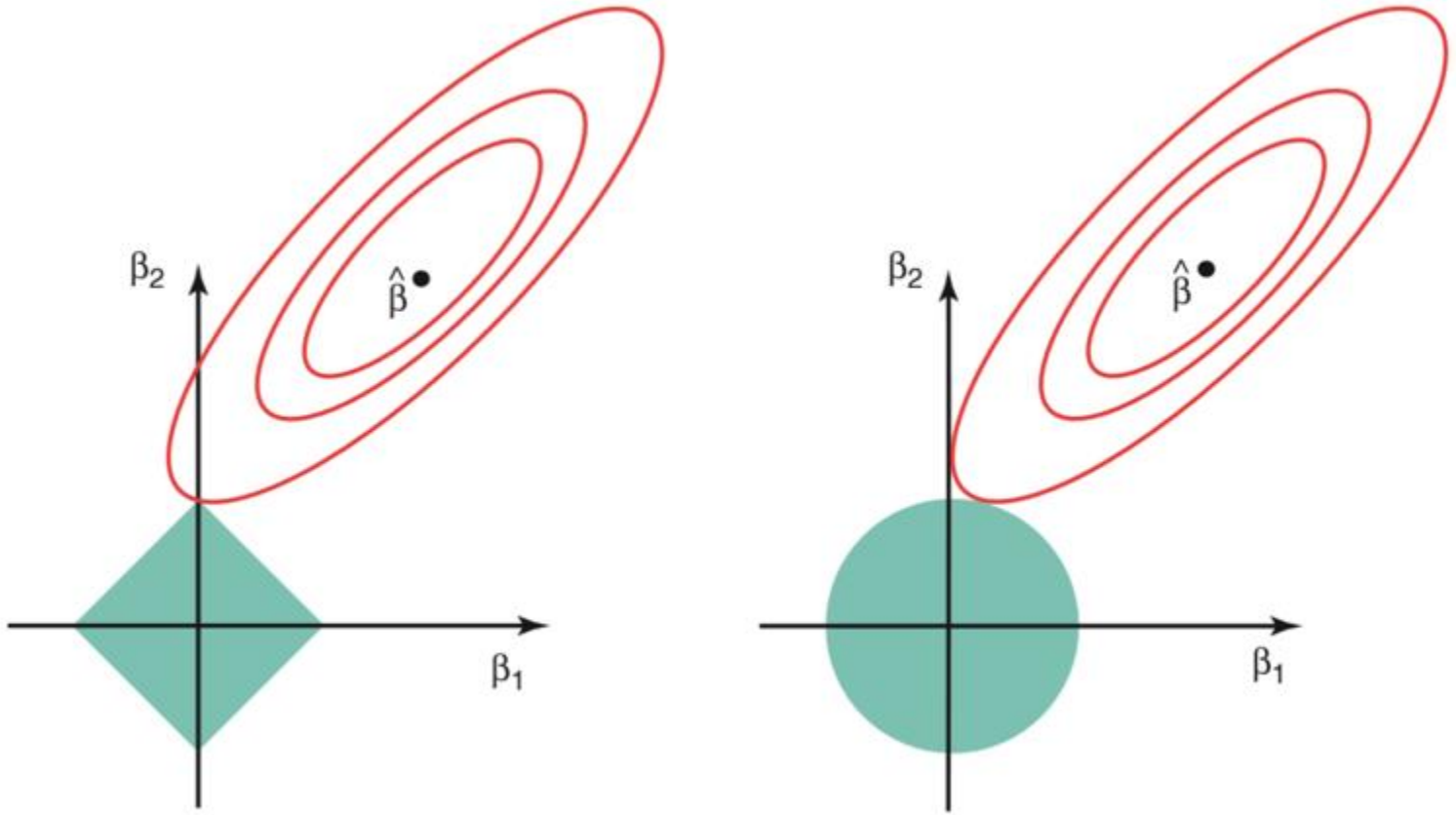
Here,

Ridge Regression : Minimize RSS Such that $\beta_1^2 + \beta_1^2 \leq S$

LASSO Regression : Minimize RSS Such that $|\beta_1| + |\beta_2| \leq S$

Regularization

We can visualize these cases as,



LASSO

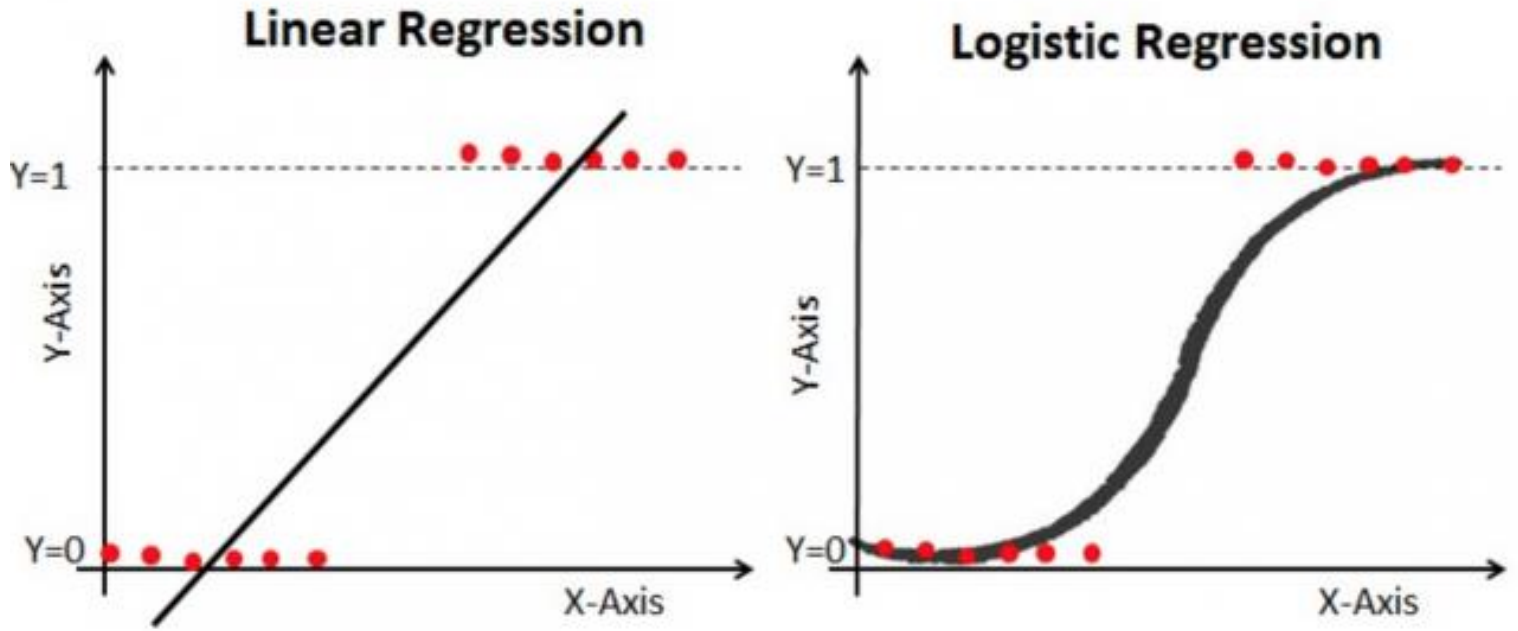
Ridge

Regularization

- Regularization, significantly reduces the variance of the model, without substantial increase in its bias.
- The tuning parameter λ, used in the regularization techniques described above, controls the impact on bias and variance.
- As the value of λ rises, it reduces the value of coefficients and thus reducing the variance.
- This controls the overfitting up to some point.
- After a some values of λ, the model starts loosing important properties, giving rise to bias in the model and thus underfitting.
- So we have to select the best λ when we are tuning this.
- There is a combined version of LASSO & Ridge. Called **Elastic Net**

Supervised Learning – Logistic Regression

Logistic regression is used when we have a dichotomous variable (Two levels categorical variable Ex- Yes/ No) as the response variable. In that case we cannot use linear regression for the predictions.



Here what we can calculate as the output of the model is the probability of belonging to the category.

Supervised Learning – Logistic Regression

Then we cannot work with this model, $y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_3 + \cdots + \beta_p x_p$. We have to convert this mode such that the output is positive as well as the output should be lying between 0 and 1. The we have to deal with the Sigmoid function.

$$S(x) = \frac{1}{1 + e^{-x}}$$

So we put above model inside to the sigmoid function and satisfy above discussed criteria.

$$P = S(y) = \frac{1}{1 + e^{-y}} = \frac{1}{1 + e^{-\beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_3 + \cdots + \beta_p x_p}}$$

So,

$$P = \frac{1}{1 + e^{-\beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_3 + \cdots + \beta_p x_p}}$$

Then we can show that,

$$log\left(\frac{P}{1 - P}\right) = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_3 + \cdots + \beta_p x_p$$

Supervised Learning – Logistic Regression

Here $log\left(\frac{P}{1-P}\right)$ is called as Logit. Parameters of the model will be estimated using Maximum Likelihood Estimation.

When predicting some variable using this model a score will be given through this model and the probability should be found and then with that probability according to some cutoff value, the prediction will be done.

Consider the example,

$$P \geq Some\ cutoff \ \longrightarrow Assign\ to\ class\ 01$$

$$P < Some\ cutoff \ \longrightarrow Assign\ to\ class\ 02$$

Generally this cutoff values is 0.5 in most cases. But this can be changed according to the situation.

Supervised Learning – Model Evaluation

The mainly discussed two techniques for evaluating a model in regression can be used for evaluating in the classification too.

- Validation Set Approach
- Cross Validation

Using these approaches we can measure some metrics for evaluating a model. The most popular metrics among these metrics is,

$$Miss\ Classificatio\ Error\ (MCE)$$

$$Accuracy = 1 - MCE$$

$$F1\ Score$$

Supervised Learning – Validation Set Approach

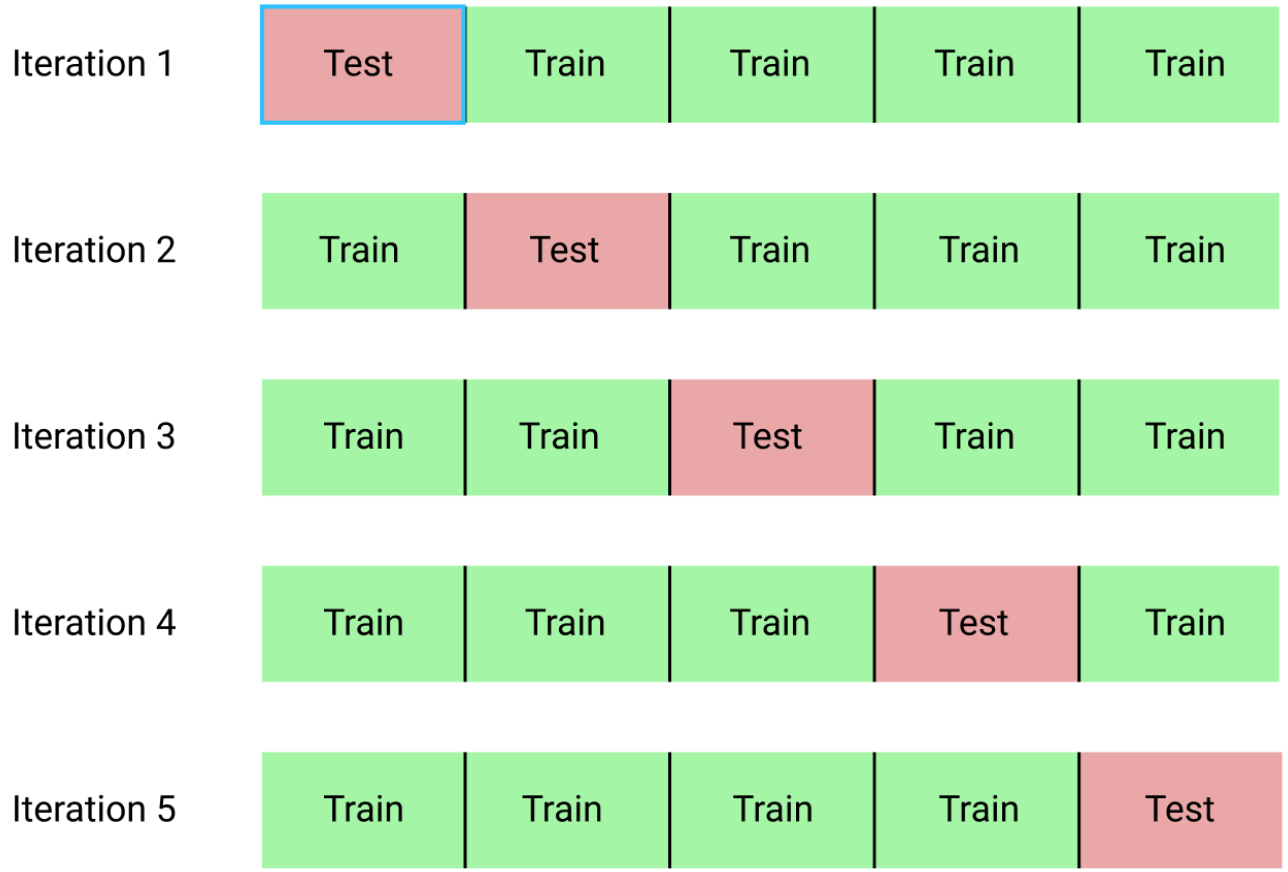
Split the dataset into two sets,

- Training dataset (Generally 80 of the data But it can be changed)
- Testing dataset (Rest of the data)

Train the model using the training dataset and then check the accuracy of the model using the testing dataset. Above mentioned metrics of a regression model will be calculated and the model will be evaluated.

Supervised Learning – Cross Validation Approach

Cross-validation, sometimes called rotation estimation or out-of-sample testing, is any of various similar model validation techniques for assessing how the results of a statistical analysis will generalize to an independent data set.



Supervised Learning – Confusion Matrix

For evaluating the above discussed metrics, a special tool is used which is called as Confusion Matrix.

Consider the following example where we have to predict a categorical variable with two levels; Yes & No. Consider that we have predicted 100 testing observations using a trained model. Then the confusion matrix can be obtained as,

	Actual Output		
Predicted Output		Yes	No
	Yes	40	10
	N0	20	30

Supervised Learning – Confusion Matrix

Correctly classified observations.

	Actual Output		
Predicted Output		Yes	No
	Yes	40	10
	N0	20	30

$$Accuracy = \frac{40 + 30}{100} \times 100\% = 70\%$$

Supervised Learning – Confusion Matrix

Incorrectly classified observations

	Actual Output		
Predicted Output		Yes	No
	Yes	40	10
	N0	20	30

$$MCE = \frac{10 + 20}{100} \times 100\% = 30\%$$

Now we can see that,

$$Acuracy = 1 - MCE$$

Supervised Learning – Confusion Matrix

		Actual Output	
		Yes	No
Predicted Output	Yes	True Positives (TP)	False Positives (FP)
	No	False Negatives (FN)	True Negatives (TN)

Precision = $\frac{\text{True Positive}}{\text{Actual Results}}$ or $\frac{\text{True Positive}}{\text{True Positive + False Positive}}$

Recall = $\frac{\text{True Positive}}{\text{Predicted Results}}$ or $\frac{\text{True Positive}}{\text{True Positive + False Negative}}$

Accuracy = $\frac{\text{True Positive + True Negative}}{\text{Total}}$

$$F_1 = \left(\frac{\text{recall}^{-1} + \text{precision}^{-1}}{2} \right)^{-1} = 2 \cdot \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}}.$$

Supervised Learning – Confusion Matrix

		Actual Output	
		Yes	No
Predicted Output	Yes	True Positives (TP)	False Positives (FP)
	No	False Negatives (FN)	True Negatives (TN)

TPR (True Positive Rate) / Recall /Sensitivity

$$\text{TPR /Recall / Sensitivity} = \frac{TP}{TP + FN}$$

Specificity

$$\text{Specificity} = \frac{TN}{TN + FP}$$

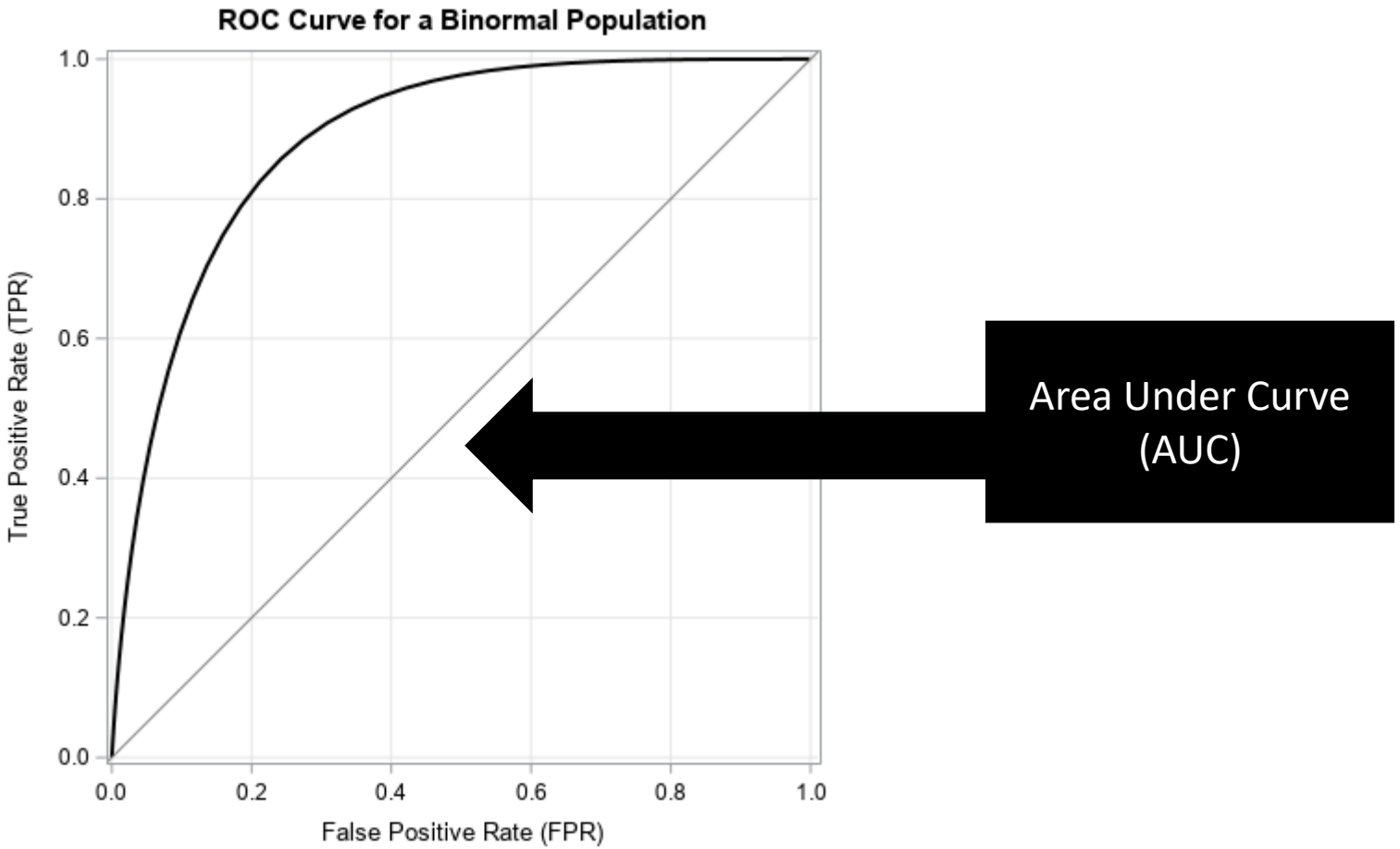
FPR

$$\text{FPR} = 1 - \text{Specificity}$$

$$= \frac{FP}{TN + FP}$$

Supervised Learning – Receiver Operating Characteristics (ROC Curve)

The ROC curve is plotted with TPR against the FPR where TPR is on the y-axis and FPR is on the x-axis when the cutoff value for classification is changed.



Supervised Learning – Area Under Curve (AUC)

With the area under the ROC curve, an idea can be got about the accuracy of a model.

AUC Range	Classification Accuracy
0.9-1.0	Excellent
0.8-0.9	Good
0.7-0.8	Fair
0.6-0.7	Bad
0.5-0.6	Very Bad

Supervised Learning – Class Imbalance Problem in Classification

Consider the following example where we have to predict Yes/ No using a model. Consider the response variable data in the training dataset.

Category	Number of Observations
Yes	1500
No	300

Here we can clearly see that the observations in both classes are not balanced. This problem is called the Class Imbalance Problem.

Supervised Learning – Class Imbalance Problem in Classification

For dealing with this problem, we have several options.

- SMOTE: Synthetic Minority Oversampling Technique
- ADASYN: Adaptive Synthetic Sampling Approach
- Hybridization: SMOTE + Tomek Links
- Hybridization: SMOTE + ENN

Supervised Learning – SMOTE: Synthetic Minority Oversampling Technique

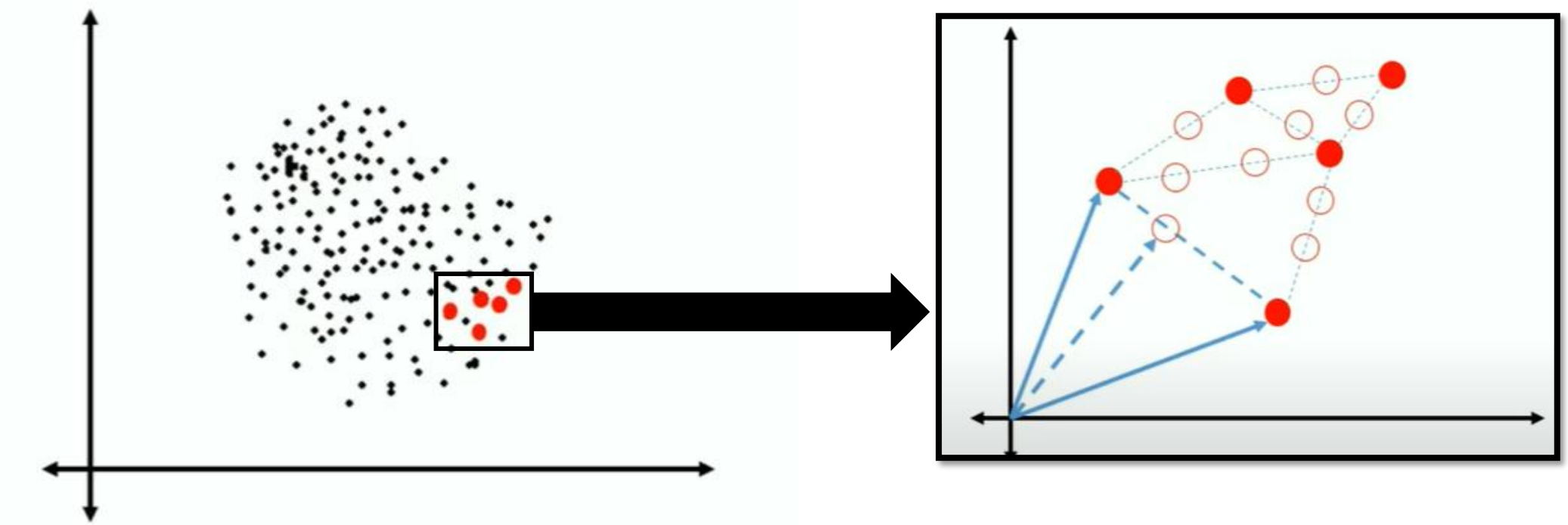
SMOTE is an oversampling technique where the synthetic samples are generated for the minority class. This algorithm helps to overcome the overfitting problem posed by random oversampling. It focuses on the feature space to generate new instances with the help of interpolation between the positive instances that lie together.

Steps:

- Identify the feature vector and its nearest neighbors
- Take the difference between two
- Multiply the difference with a random number between 0 and 1
- Identify a new point on the line segment by adding the random number to feature vector
- Repeat the process for identified feature vectors

Supervised Learning – SMOTE: Synthetic Minority Oversampling Technique

Consider the following example.



Two Class Classification

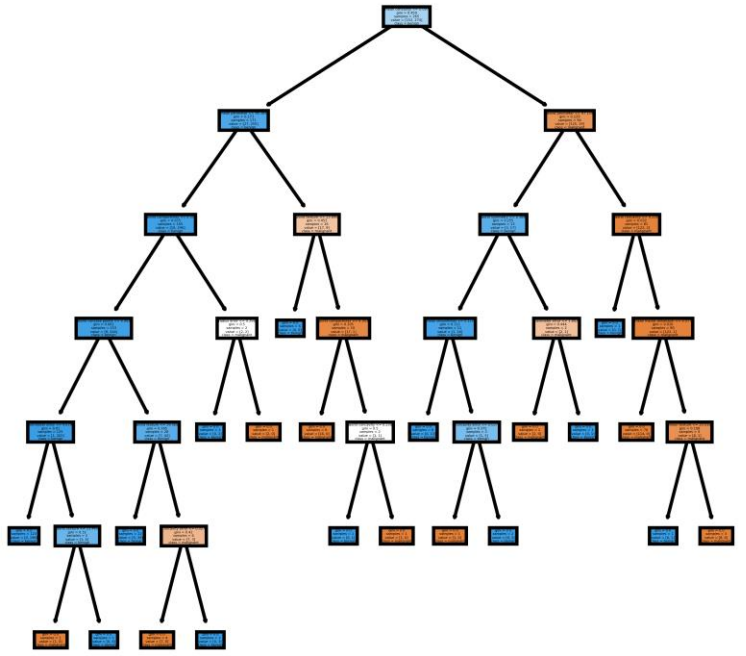
No-Fraud → 99.5%
Fraud → 0.5%

Supervised Learning – Classification & Regression Trees (CART)

A Decision Tree (CART) is a simple representation for classification as well as for regression. It is a Supervised Machine Learning where the data is continuously split according to a certain parameter.

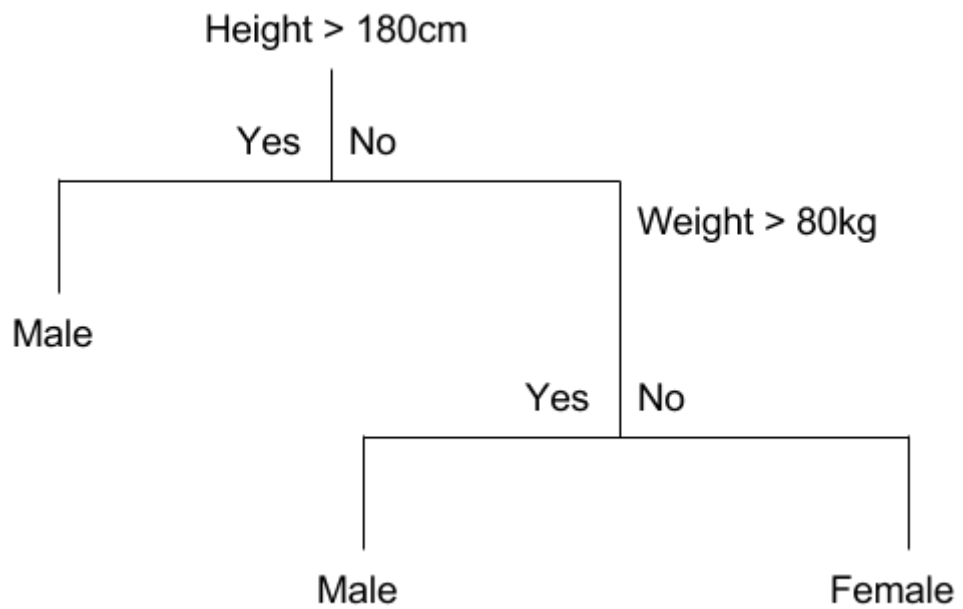
There are two types of Decision Trees we can identify.

- Classification Trees
- Regression Trees



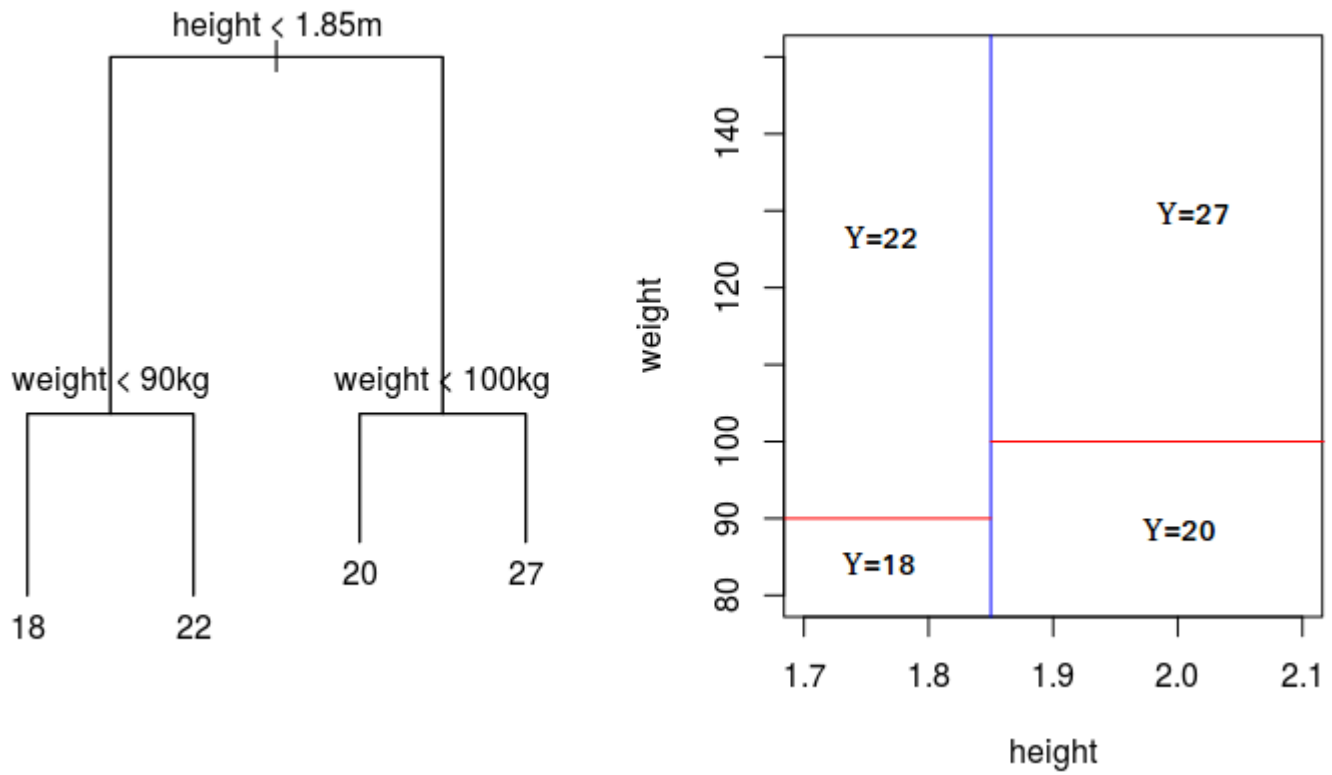
Supervised Learning – Classification Trees

Here the response variable is a categorical variable. So the main objective is to classify observations. Consider the following example.



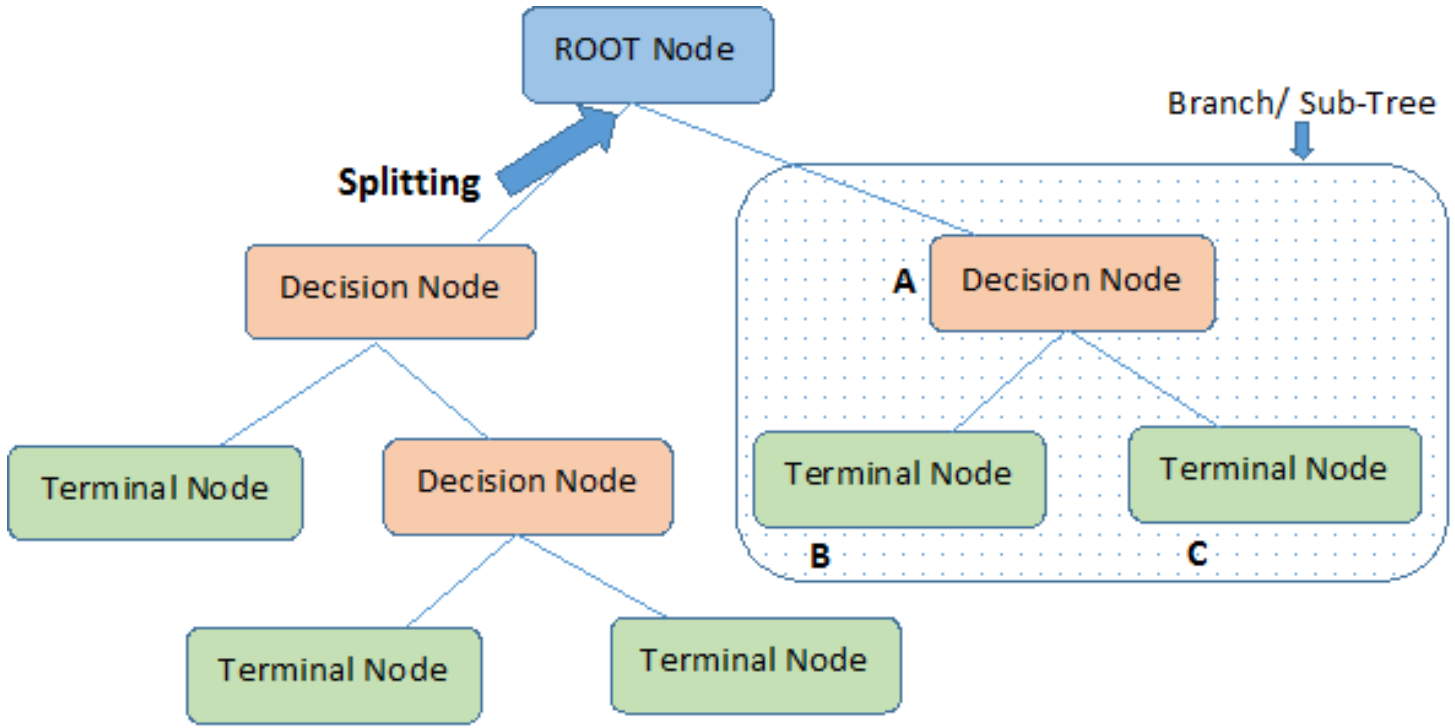
Supervised Learning – Regression Trees

Here the response variable is a numerical variable. So the main objective is to predict observations. Consider the following example. Here the average value is returned such that the criteria is satisfied.



Supervised Learning – Important Terminologies Related to Decision Trees

Consider the following diagram. All the terminologies are given.



Note:- A is parent node of B and C.

Supervised Learning – Splitting in Decision Trees

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

There are several techniques we can use to do this.

- Gini Impurity
- Information Gain
- Reduction in Variance

Supervised Learning – Gini Impurity

This says that if we select two items from a population at random then they must be of same class and probability for this is 1 if population is pure.

- It works with categorical target variable “Success” or “Failure”.
- It performs only Binary splits
- Higher the value of Gini higher the homogeneity.
- CART (Classification and Regression Tree) uses Gini method to create binary splits.

Steps to Calculate Gini for a split

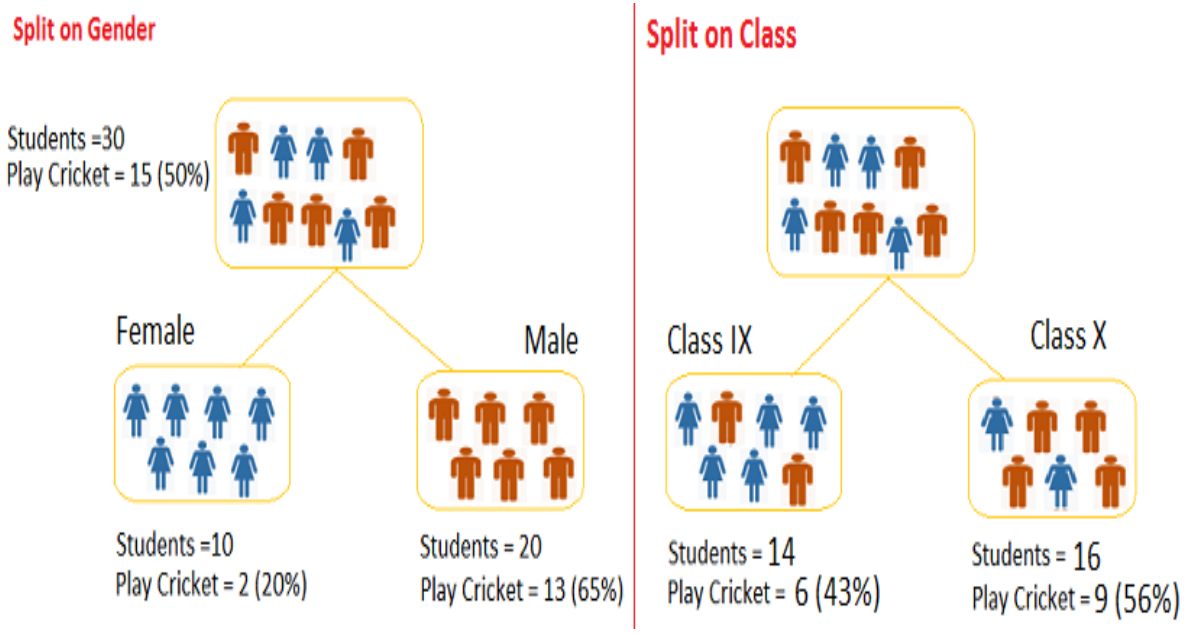
- Calculate Gini for sub-nodes, using formula sum of square of probability for success and failure

$$p^2 + (1 - p)^2$$

- Calculate Gini for split using weighted Gini score of each node of that split.

Supervised Learning – Gini Impurity

Consider the following example. Here we want to segregate the students based on target variable (playing cricket or not). In the snapshot below, we split the population using two input variables Gender and Class. The objective is to identify most homogeneous subgroups.



Split on Gender:
Calculate, Gini for sub-node Female = $(0.2)^2+(0.8)^2= 0.68$
Gini for sub-node Male = $(0.65)^2+(0.35)^2= 0.55$
Calculate weighted Gini for Split Gender = $(10/30) \times 0.68 + (20/30) \times 0.55 = \mathbf{0.59}$

Similar for Split on Class:
Gini for sub-node Class IX = $(0.43)^2+(0.57)^2= 0.51$
Gini for sub-node Class X = $(0.56)^2+(0.44)^2= 0.51$
Calculate weighted Gini for Split Class = $(14/30) \times 0.51 + (16/30) \times 0.51 = \mathbf{0.51}$

Supervised Learning – Gini Impurity

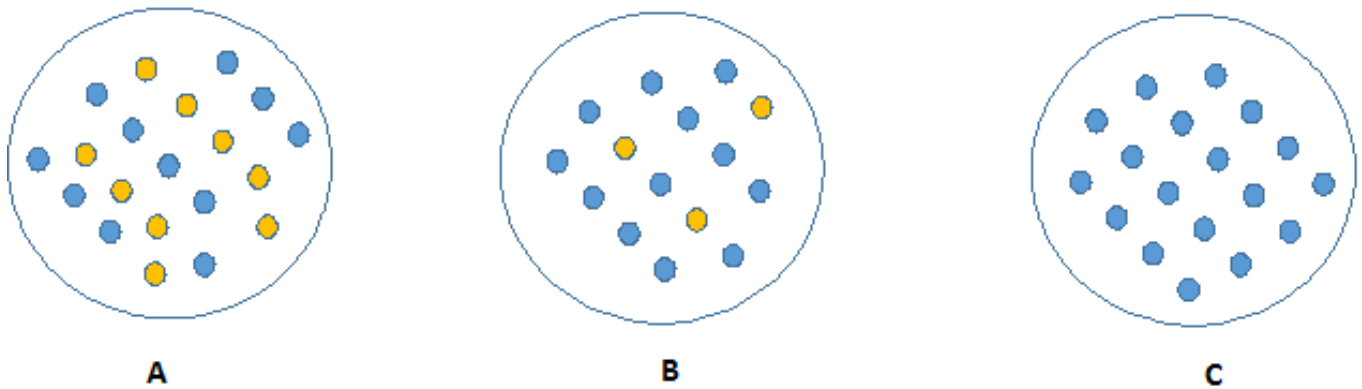
We can see that Gini score for Split on Gender is higher than Split on Class, hence, the node split will take place on Gender. We can work with the Gini Impurity value as well. It is nothing but,

$$Gini\ Impurity\ =\ 1 - Gini$$

So if we go with the Gini Impurity, we select the lowest Gini Impurity. In that case also, here we have to select Gender.

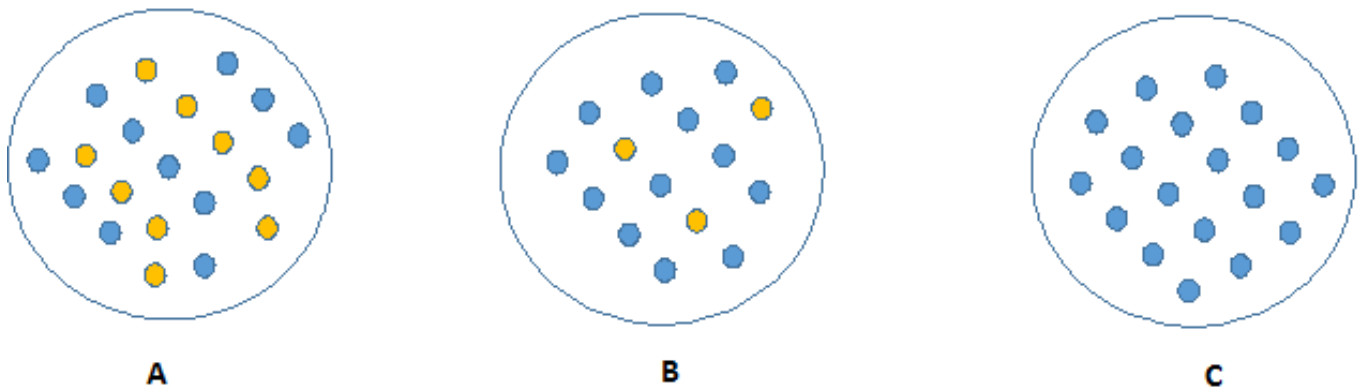
Supervised Learning – Information Gain

Think which can be described easily among following groups.



Supervised Learning – Information Gain

Think which can be described easily among following groups.



Answer is C because it requires less information as all values are similar. On the other hand, B requires more information to describe it and A requires the maximum information. In other words, we can say that C is a Pure node, B is less impure, and A is more impure.

Supervised Learning – Information Gain

we can build a conclusion that less impure node requires less information to describe it. And, more impure node requires more information. Information theory is a measure to define this degree of disorganization in a system known as Entropy. If the sample is completely homogeneous, then the entropy is zero and if the sample is an equally divided (50% – 50%), it has entropy of one.

Entropy can be calculated using formula:-

$$Entropy = -plog_2(p) - (1-p)log_2(1-p)$$

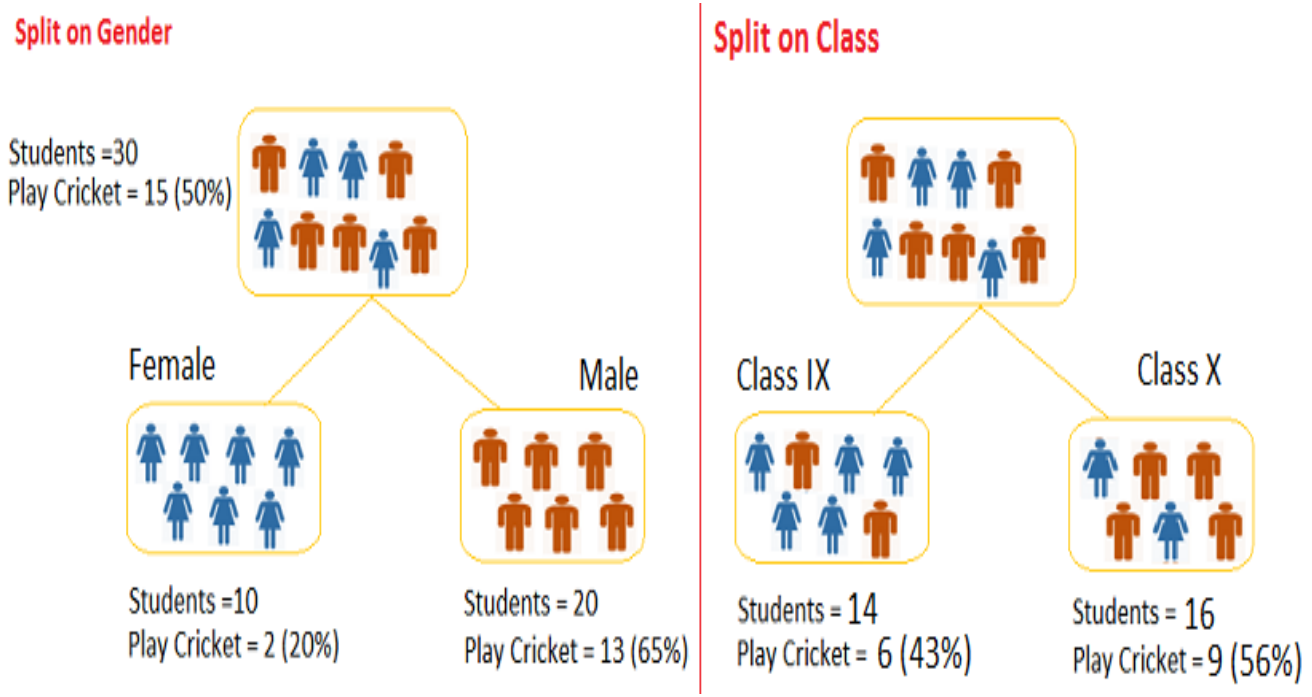
Here the p is the probability of success. The lesser the entropy, the better it is.

Steps to calculate entropy for a split:

- Calculate entropy of parent node
- Calculate entropy of each individual node of split and calculate weighted average of all sub-nodes available in split.

Supervised Learning – Information Gain

Consider the above example



Entropy for parent node = -(15/30) log2 (15/30) – (15/30) log2 (15/30) = **1**. Here 1 shows that it is an impure node.

Entropy for Female node = -(2/10) log2 (2/10) – (8/10) log2 (8/10) = 0.72 and for male node, -(13/20) log2 (13/20) – (7/20) log2 (7/20) = 0.93
Entropy for split Gender = Weighted entropy of sub-nodes = (10/30)*0.72 + (20/30)*0.93 = **0.86**

Entropy for Class IX node, -(6/14) log2 (6/14) – (8/14) log2 (8/14) = 0.99 and for Class X node, -(9/16) log2 (9/16) – (7/16) log2 (7/16) = 0.99.
Entropy for split Class = (14/30)*0.99 + (16/30)*0.99 = **0.99**

Supervised Learning – Information Gain

Information gain is considered here as,

$$Information\ Gain = Entropy(Parent) - Entropy(Split)$$

Maximum information gain is given in the lowest Entropy. Entropy for the split of the Gender is the lowest. So select the split with Gender.

Supervised Learning – Reduction in Variance

Reduction in variance is an algorithm used for continuous target variables (regression problems). This algorithm uses the standard formula of variance to choose the best split. The split with lower variance is selected as the criteria to split the population:

$$Variance = \frac{\sum_{i=1}^{n}(x_i - \bar{x})^2}{n}$$

$\bar{x}$ is the mean of the values.

Steps to calculate Variance:

- Calculate variance for each node.
- Calculate variance for each split as weighted average of each node variance.

Supervised Learning – Reduction in Variance

Consider the above example. Let's assign numerical value 1 for play cricket and 0 for not playing cricket.

- Variance for Root node, here mean value is $(15 \times 1 + 15 \times 0)/30 = 0.5$ and we have 15 one and 15 zero.
- Now variance would be $(15 \times (1 - 0.5)\text{^}2 + 15 \times (0 - 0.5)\text{^}2) / 30 = \mathbf{0.25}$

- Mean of Female node = $(2 \times 1 + 8 \times 0)/10 = 0.2$ and Variance = $(2 \times (1 - 0.2)^2 + 8 \times (0 - 0.2)^2) / 10 = 0.16$
- Mean of Male Node = $(13 \times 1 + 7 \times 0)/20 = 0.65$ and Variance = $(13 \times (1 - 0.65)^2 + 7 \times (0 - 0.65)^2) / 20 = 0.23$
- Variance for Split Gender = Weighted Variance of Sub-nodes = $(10/30) \times 0.16 + (20/30) \times 0.23 = \mathbf{0.21}$

- Mean of Class IX node = $(6 \times 1 + 8 \times 0)/14 = 0.43$ and Variance = $(6 \times (1 - 0.43)^2 + 8 \times (0 - 0.43)^2) / 14 = 0.24$
- Mean of Class X node = $(9 \times 1 + 7 \times 0)/16 = 0.56$ and Variance = $(9 \times (1 - 0.56)^2 + 7 \times (0 - 0.56)^2 / 16 = 0.25$
- Variance for Split Class = $(14/30) \times 0.24 + (16/30) \times 0.25 = \mathbf{0.25}$

We can see that Gender split has lower variance compare to parent node, so the split would take place on Gender variable.

Supervised Learning – Advantages & Disadvantages

Advantages

- Easy to Understand
- Useful in Data exploration
- Less data cleaning required
- Data type is not a constraint
- Non-Parametric Method

Disadvantages

- Over fitting

Supervised Learning – Pruning

Overfitting is one of the key challenges faced while using tree based algorithms. If there is no limit set of a decision tree, it will give you 100% accuracy on training set because in the worse case it will end up making 1 leaf for each observation. Thus, preventing overfitting is pivotal while modeling a decision tree. This can be done through Pruning trees.

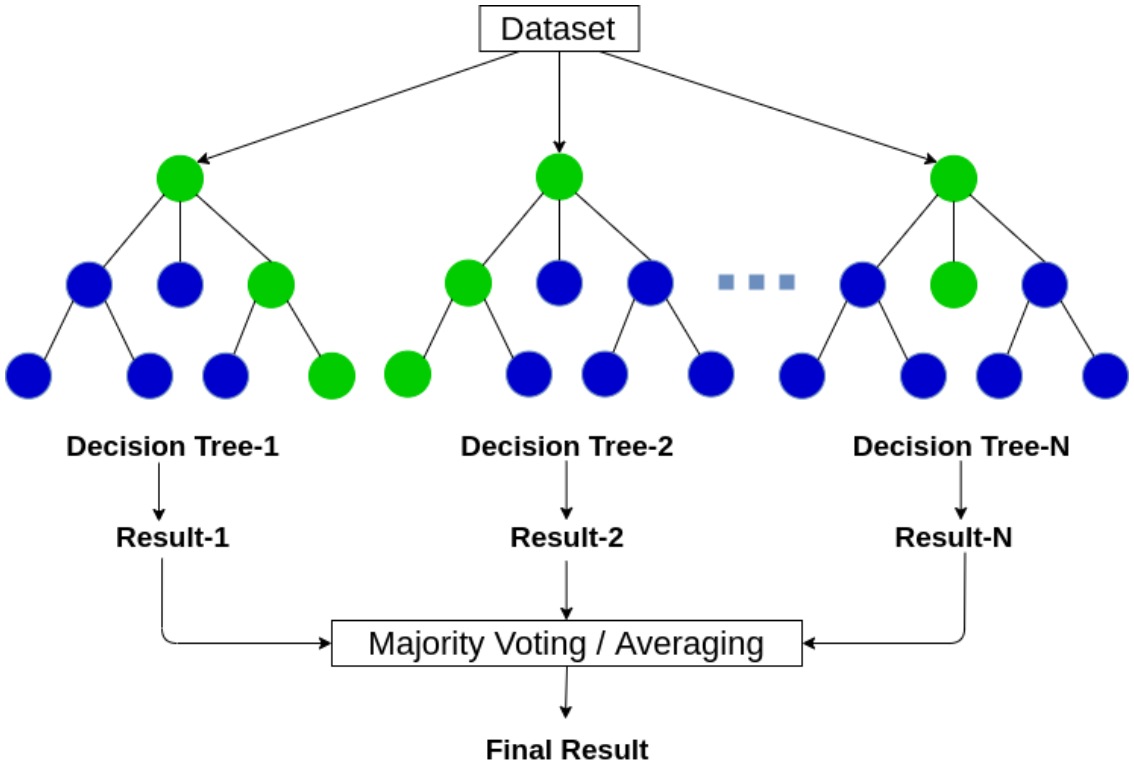
Techniques for reducing the complexity of trees.

- Maximum depth of tree
- Maximum number of terminal nodes
- Maximum features to consider for split

These are several techniques. There are much more. Another way of reducing this overfitting is **Random Forest** algorithm.

Supervised Learning – Random Forest

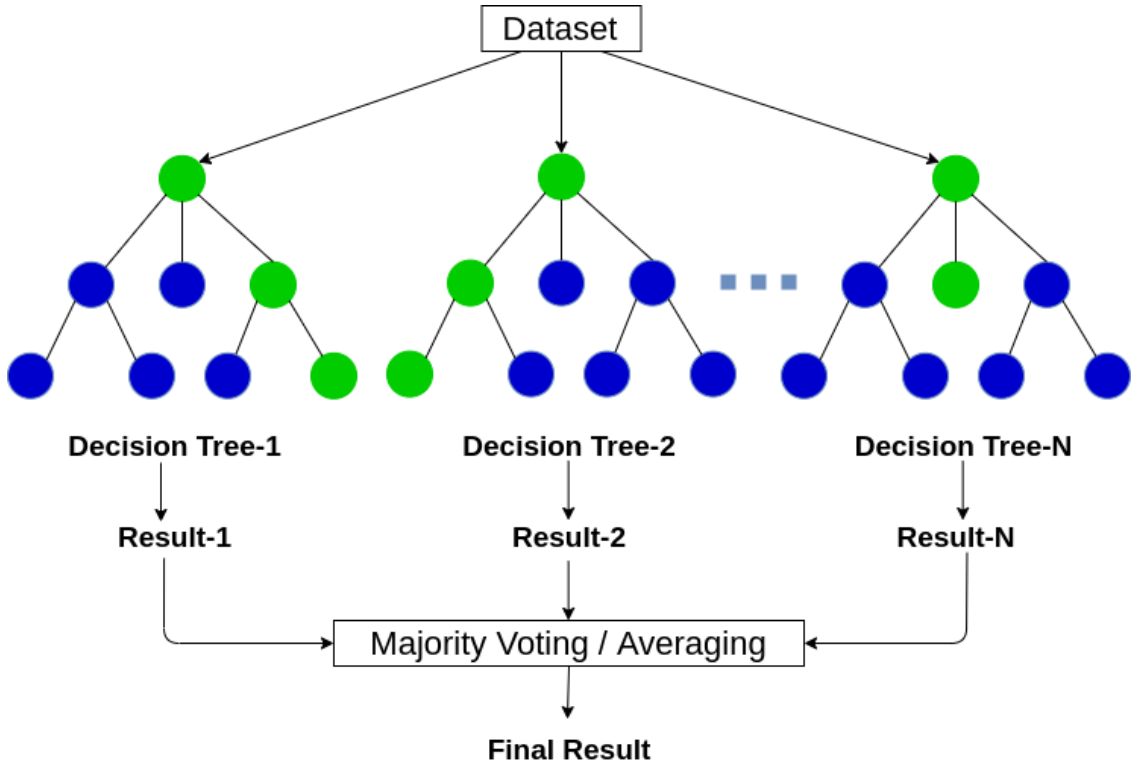
Here several decision trees are created using bootstrapped random samples from the original training dataset and the predictions are made using the outcomes of all the trees.



For classification majority vote will be taken and for the regression the average of all the trees will be taken. Another importance is in each split of each tree, a subset of variables will be taken which are highly important for splitting. Normally $\sqrt{P}$ of variables are usen in each split when there are P number of variables available in the original data.

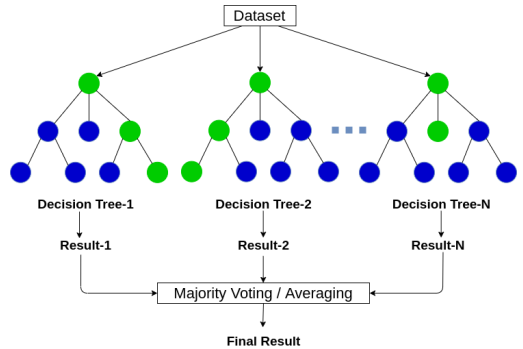
Supervised Learning – Extremely Randomized Trees

Here several decision trees are created using the original training dataset and the predictions are made using the outcomes of all the trees.



For classification majority vote will be taken and for the regression the average of all the trees will be taken. Another importance is in each split of each tree, a random set of variables will be taken for splitting. Normally $\sqrt{P}$ of variables are usen in each split when there are P number of variables available in the original data.

Supervised Learning – Random Forest VS Extremely Randomized Trees



	Decision Tree	Random Forest	Extra Trees
Number of trees	1	Many	Many
No of features considered for split at each decision node	All Features	Random subset of Features	Random subset of Features
Boostrapping(Drawing Sampling without replacement)	Not applied	Yes	No
How split is made	Best Split	Best Split	Random Split

Supervised Learning – K- Nearest Neighbors

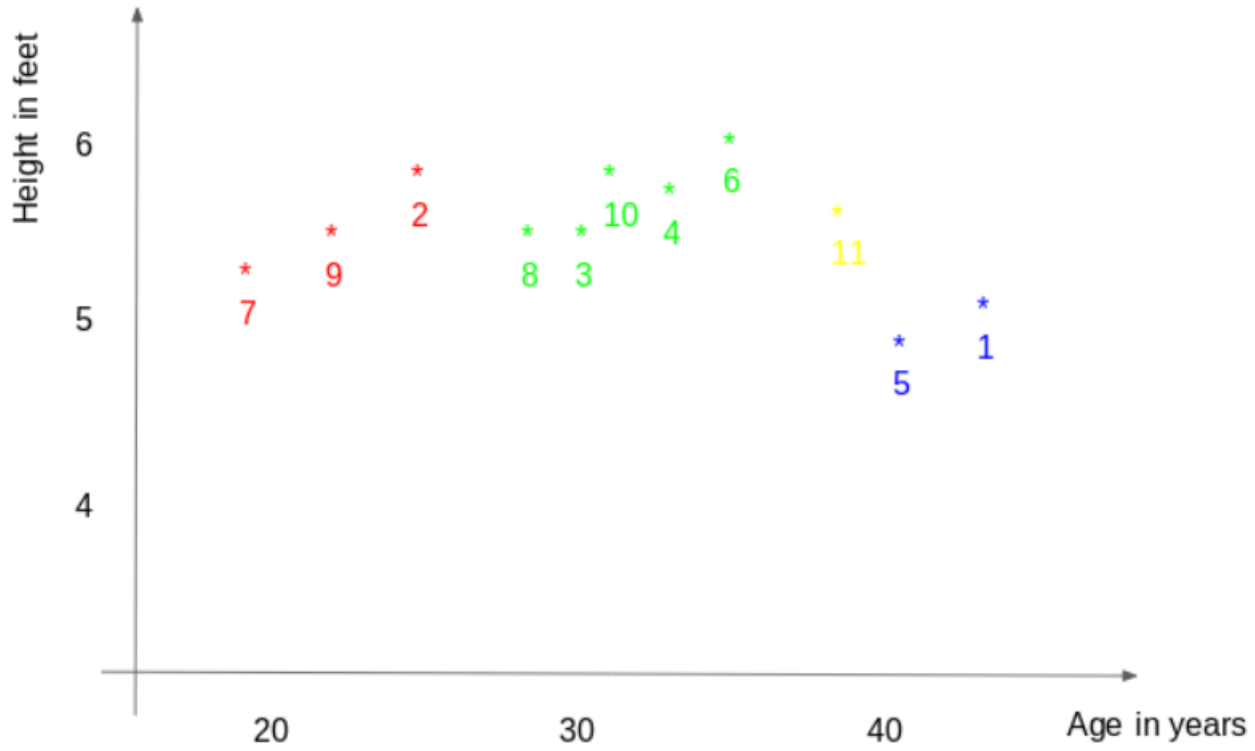
Consider the following example. Think that we are going to fit a model using Height and Age for the response variable Weight.

ID	Height	Age	Weight
1	5	45	77
2	5.11	26	47
3	5.6	30	55
4	5.9	34	59
5	4.8	40	72
6	5.8	36	60
7	5.3	19	40
8	5.8	28	60
9	5.5	23	45
10	5.6	32	58
11	5.5	38	?

Think that we need to find the Weight for observation 11 using the given data.

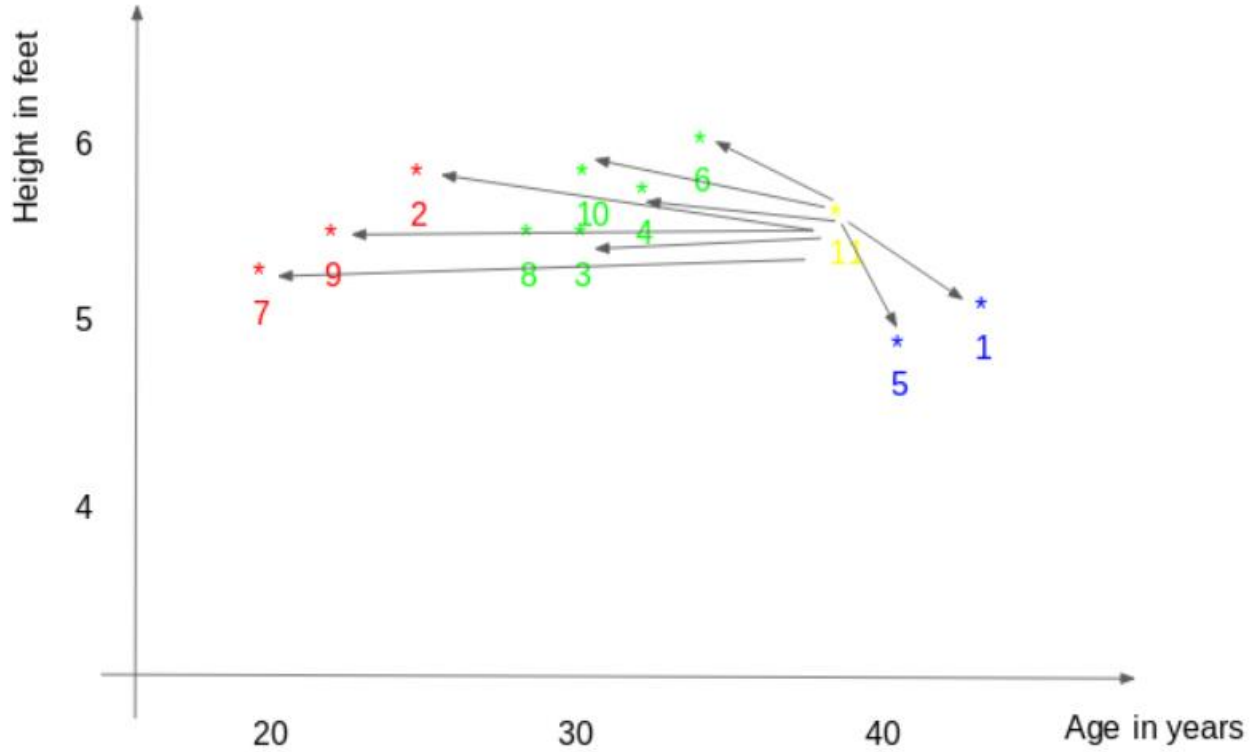
Supervised Learning – K- Nearest Neighbors

If we plot this,



Supervised Learning – K- Nearest Neighbors

First, the distance between the new point and each training point is calculated.



Supervised Learning – K- Nearest Neighbors

For this distance calculations we can use 3 types of distances.

- Euclidean Distance: Euclidean distance is calculated as the square root of the sum of the squared differences between a new point (x) and an existing point (y).

$$\sqrt{\sum_{i=1}^{k}(x_i - y_i)^2}$$

- Manhattan Distance: This is the distance between real vectors using the sum of their absolute difference.

$$\sum_{i=1}^{k}|x_i - y_i|$$

- Hamming Distance: It is used for categorical variables. If the value (x) and the value (y) are the same, the distance D will be equal to 0 . Otherwise D=1.

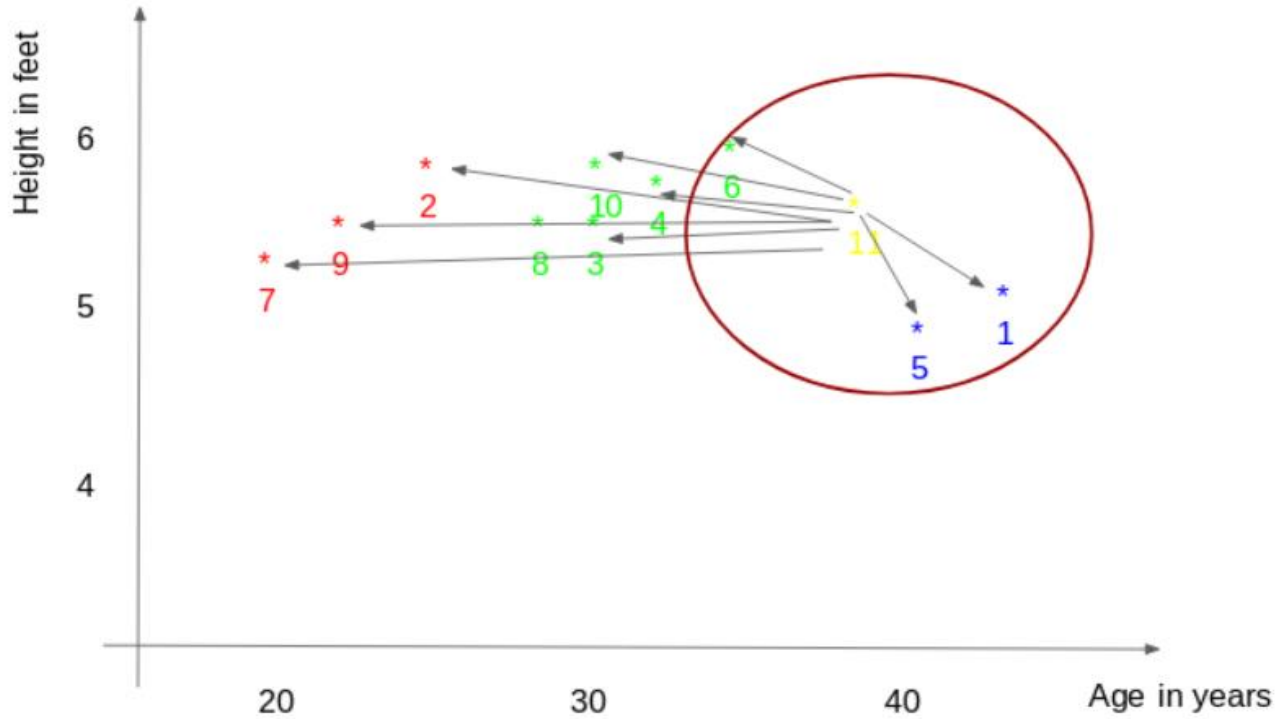
$$D_H = \sum_{i=1}^{k}|x_i - y_i|$$

$$x = y \Rightarrow D = 0$$

$$x \neq y \Rightarrow D = 1$$

Supervised Learning – K- Nearest Neighbors

The closest k data points are selected (based on the distance). Consider here k=3, so in this example, points 1, 5, 6 will be selected if the value of k is 3.



Then the average of these data points is the final prediction for the new point. Here, we have weight of ID11 is (77+72+60)/3 = 69.66 kg. If we have a classification case, the most frequent category will be chosen to assign.

Supervised Learning – K- Nearest Neighbors

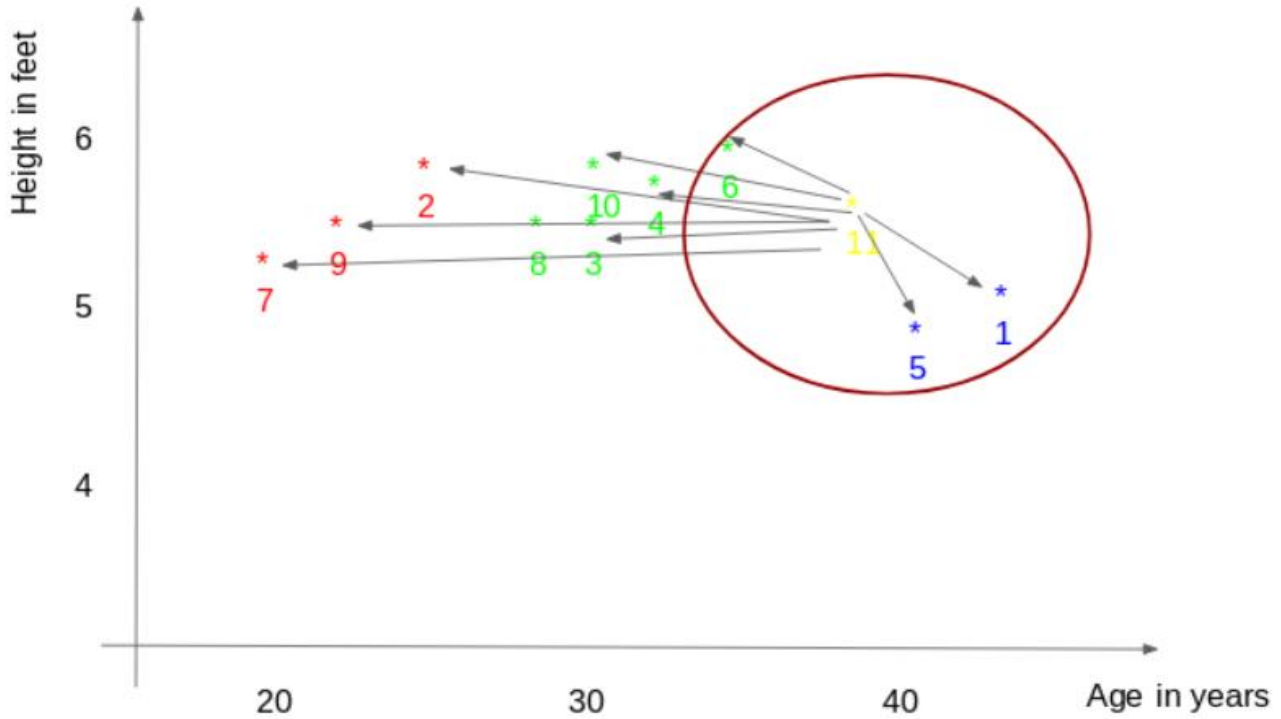
Consider the following example. Think that we are going to fit a model using Height and Age for the response variable Weight.

ID	Height	Age	Weight	Class
1	5	45	77	A
2	5.11	26	47	A
3	5.6	30	55	B
4	5.9	34	59	C
5	4.8	40	72	B
6	5.8	36	60	A
7	5.3	19	40	C
8	5.8	28	60	A
9	5.5	23	45	B
10	5.6	32	58	C
11	5.5	38	?	

Think that we need to find the Class for observation 11 using the given data.

Supervised Learning – K- Nearest Neighbors

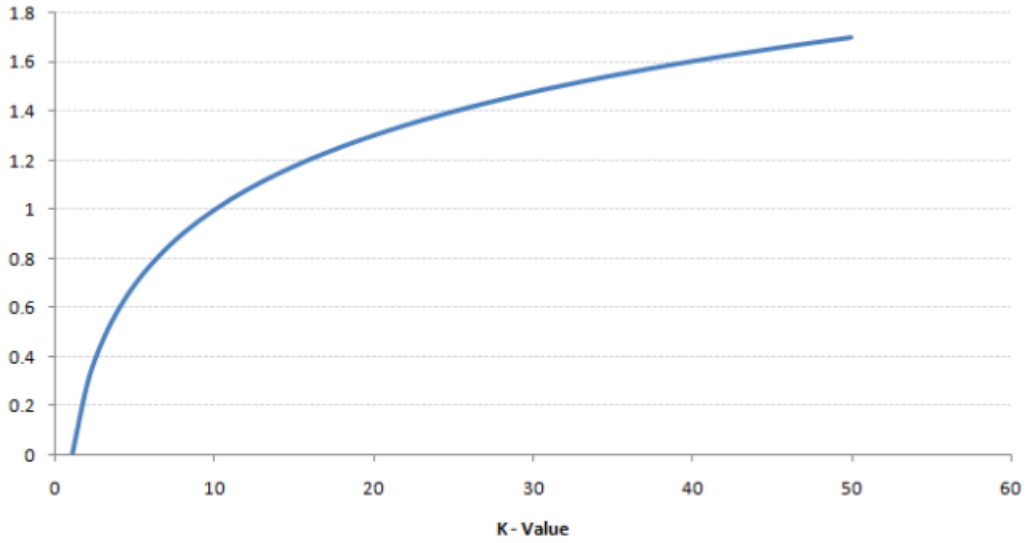
The closest k data points are selected (based on the distance). Consider here k=3, so in this example, points 1, 5, 6 will be selected if the value of k is 3.

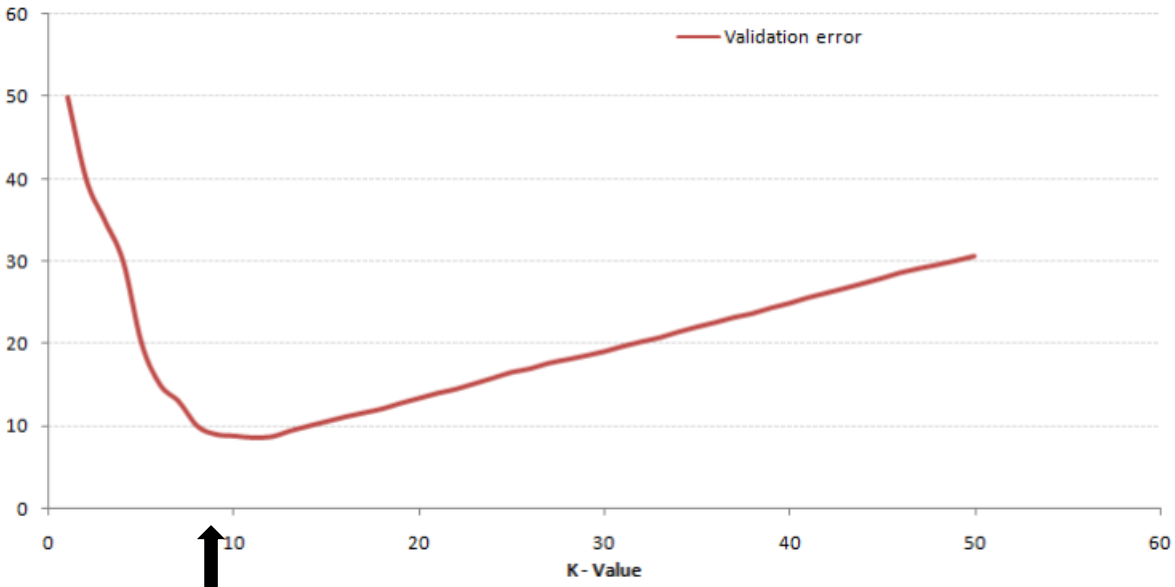


If we have a classification case, the most frequent category will be chosen to assign. 11th observation belongs to Class A.

Supervised Learning – K- Nearest Neighbors

To select the optimum k for the algorithm, we can use the Validation Set or Cross Validation approaches. Consider following graphs. Training & validation errors have been measured for several k values.

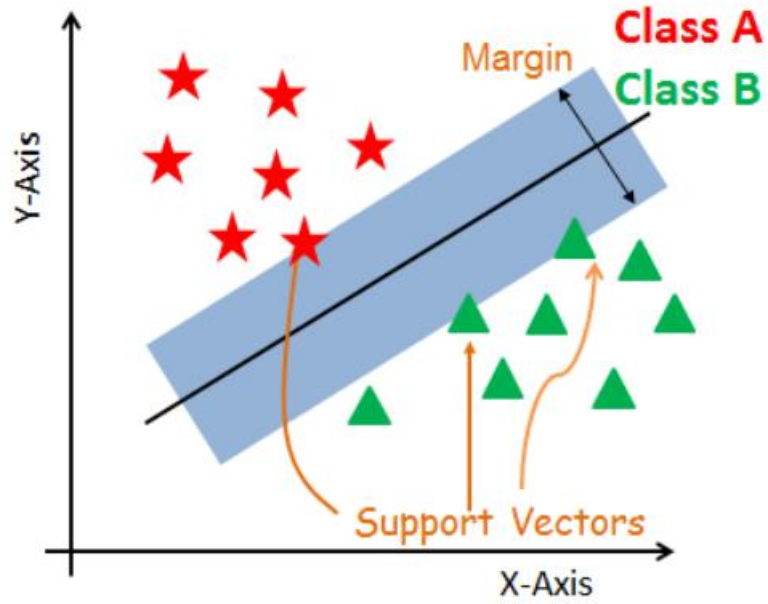




k=9

Supervised Learning – Support Vector Machines

SVM constructs a hyperplane in multidimensional space to separate different classes. SVM generates optimal hyperplane in an iterative manner, which is used to minimize an error. The core idea of SVM is to find a maximum marginal hyperplane(MMH) that best divides the dataset into classes.

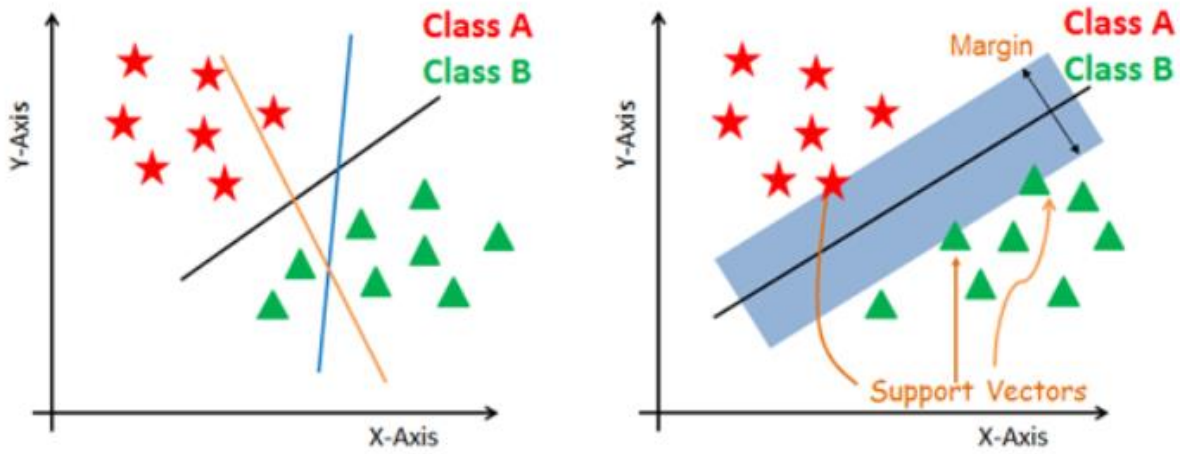


This is generally used for classification. But for regression, this can be extended.

Supervised Learning – Support Vector Machines

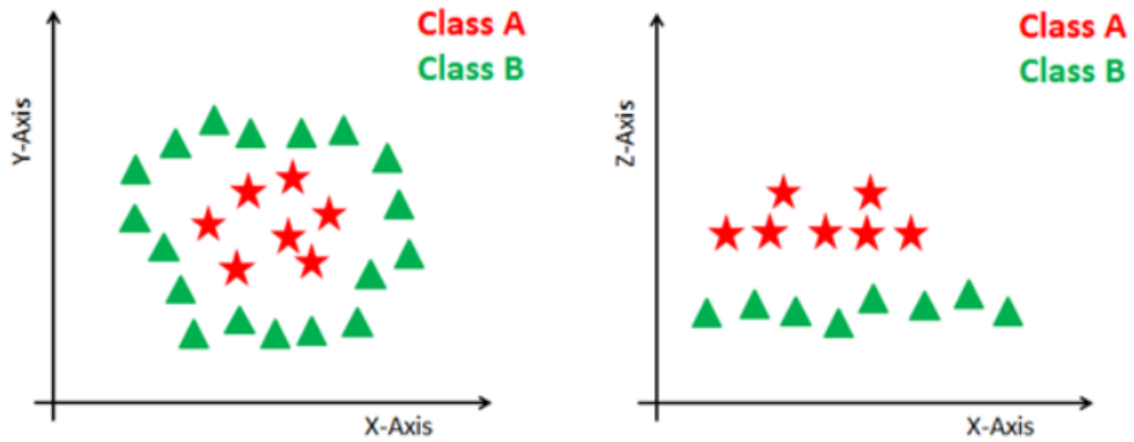
SVM searches for the maximum marginal hyperplane in the following steps:

1. Generate hyperplanes which segregates the classes in the best way. Left-hand side figure showing three hyperplanes black, blue and orange. Here, the blue and orange have higher classification error, but the black is separating the two classes correctly.
2. Select the right hyperplane with the maximum segregation from the either nearest data points as shown in the right-hand side figure.



Supervised Learning – Support Vector Machines

Some problems can’t be solved using linear hyperplane, as shown in the figure below .In such situation, SVM uses a kernel trick to transform the input space to a higher dimensional space as shown on the right.



Some popular kernels are,

- Linear Kernel
- Polynomial Kernel
- Radial Basis Function Kernel

Supervised Learning – Hyper Parameter Optimization

What are hyperparameters?

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

In machine learning, hyperparameter optimization or tuning is the problem of choosing a set of optimal hyperparameters for a learning algorithm. There are two popular techniques for satisfying this objective.

- Grid Search Algorithm
- Random Search Algorithm

Search Space: Volume to be searched where each dimension represents a hyperparameter and each point represents one model configuration.

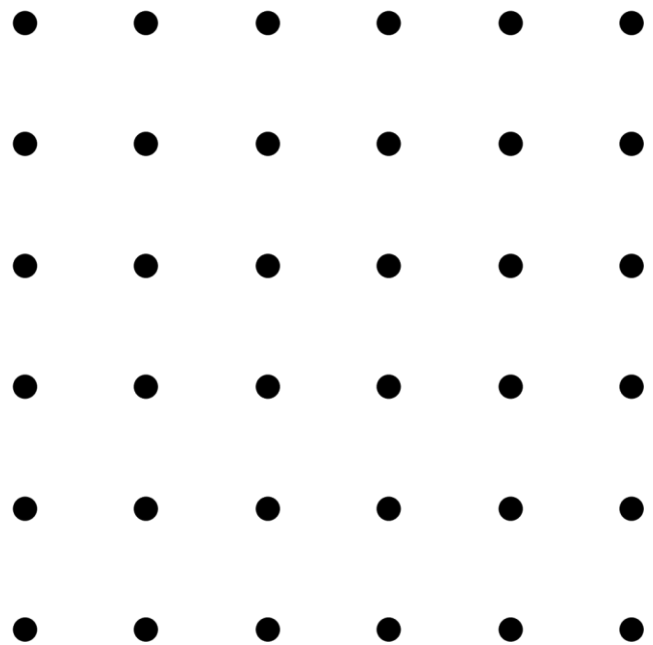
Random Search: Define a search space as a bounded domain of hyperparameter values and randomly sample points in that domain.

Grid Search: Define a search space as a grid of hyperparameter values and evaluate every position in the grid.

Supervised Learning – Hyper Parameter Optimization

Grid Search

Grid Search can be thought of as an exhaustive search for selecting a model. In Grid Search, the data scientist sets up a grid of hyperparameter values and for each combination, trains a model and scores on the testing data. In this approach, every combination of hyperparameter values is tried



Supervised Learning – Hyper Parameter Optimization

Random Search

Random Search sets up a grid of hyperparameter values and selects random combinations to train the model and score. This allows you to explicitly control the number of parameter combinations that are attempted. The number of search iterations is set based on time or resources.

Unsupervised Learning – K- Means Clustering

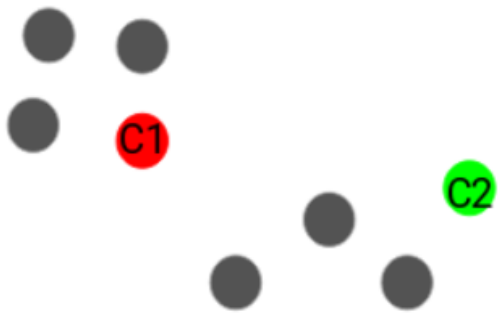
A K-means clustering algorithm tries to group similar items in the form of clusters. The number of groups is represented by K.

Consider the following data points.

Unsupervised Learning – K- Means Clustering

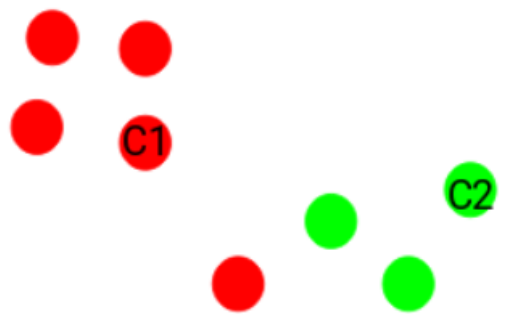
Step 1: Choose the number of clusters k

Step 2: Select k random points from the data as centroids



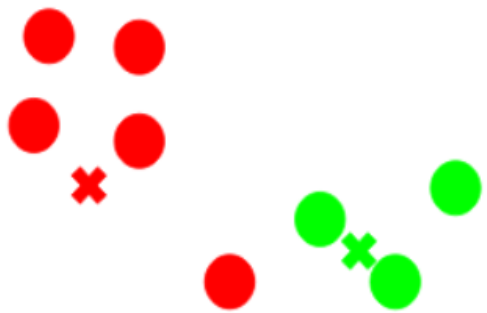
Unsupervised Learning – K- Means Clustering

Step 3: Assign all the points to the closest cluster centroid



Unsupervised Learning – K- Means Clustering

Step 4: Recompute the centroids of newly formed clusters



Unsupervised Learning – K- Means Clustering

Step 5: Repeat steps 3 and 4

Unsupervised Learning – K- Means Clustering

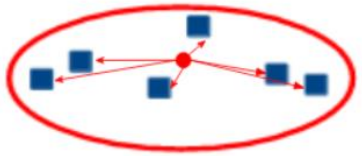
There are essentially three stopping criteria that can be adopted to stop the K-means algorithm:

- Centroids of newly formed clusters do not change
- Points remain in the same cluster
- Maximum number of iterations are reached

Unsupervised Learning – Evaluating Clusters

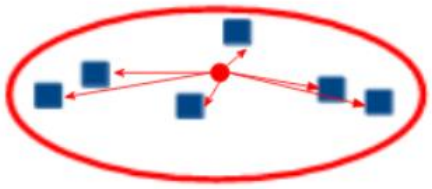
There are no many metrics which can be used for measuring the accuracy of the unsupervised learning algorithms as the supervised learning. But two metrics can be used for evaluating the clusters.

- Inertia : Calculates the sum of distances of all the points within a cluster from the centroid of that cluster.

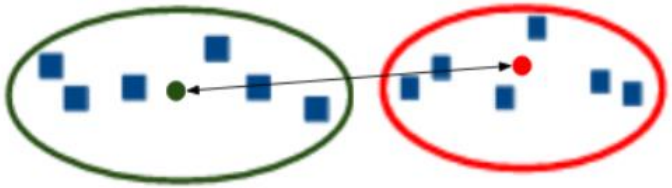


Intra cluster distance

- Dunn Index: Takes into account the distance between two clusters.



Intra cluster distance

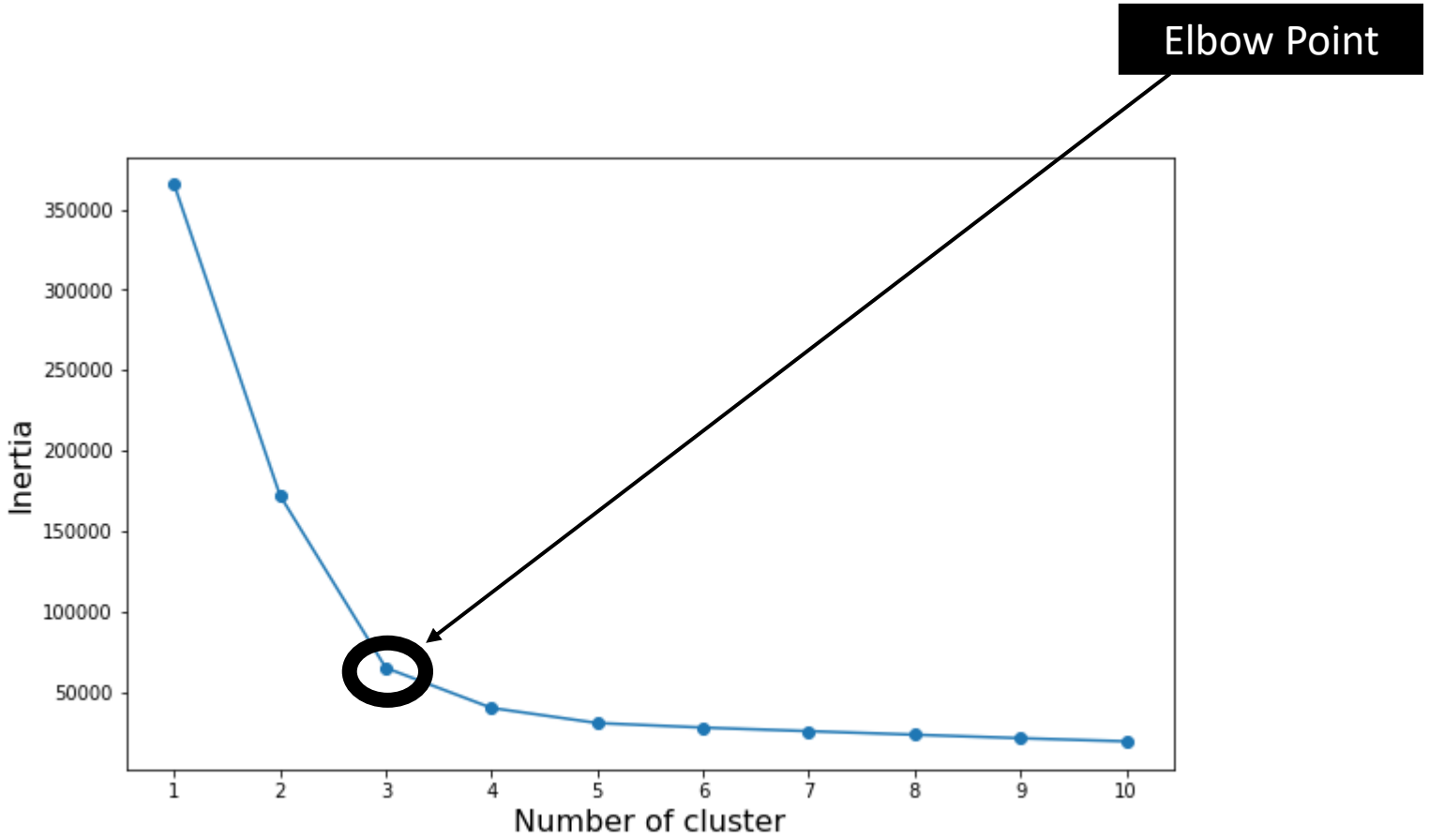


Inter cluster distance

$$\text{Dunn Index} = \frac{\min(\text{Inter cluster distance})}{\max(\text{Intra cluster distance})}$$

Unsupervised Learning – Selecting K in K- Means Clustering

Number of clusters or K can be found such that the inertia is minimized.

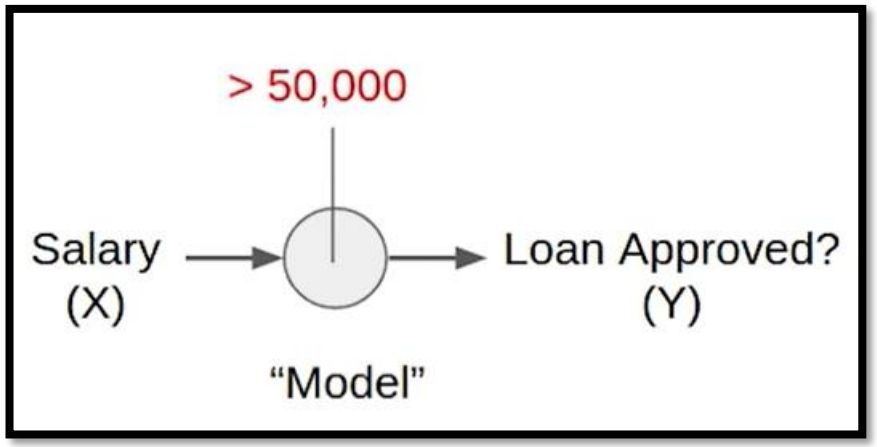


Deep Learning – Perceptron

Think about following question. Here with the salary, it will check whether the loan is approved or not.

Deep Learning – Perceptron

Think that the loan is going to approved when the salary is greater than 50000.

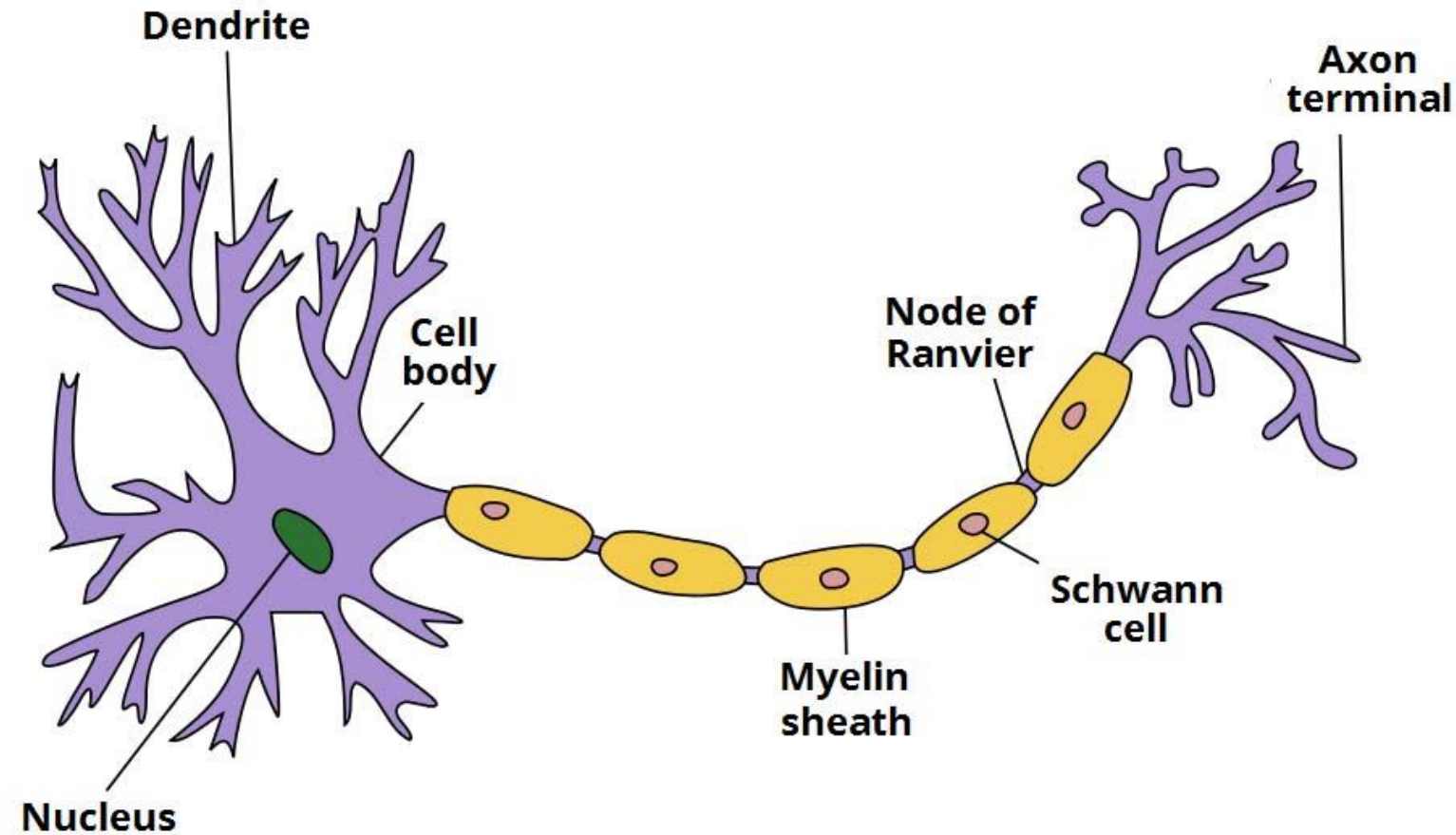


Tasks:

- Salary as input
- Check if it is at least 50000 or not?
- If the condition is True the approve the loan

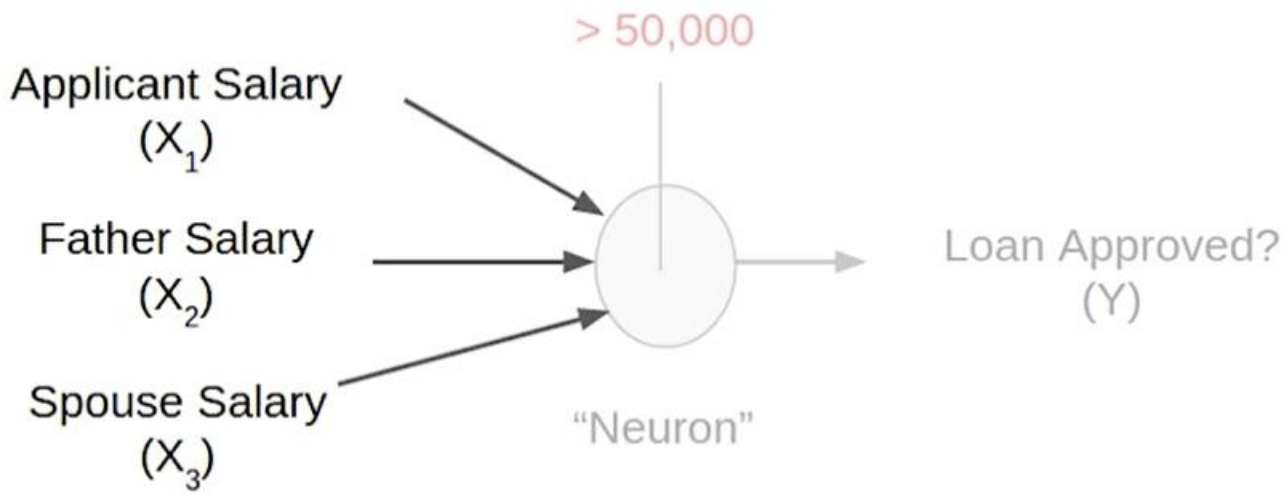
Deep Learning – Perceptron

This the general process of decision making inside a biological neuron.



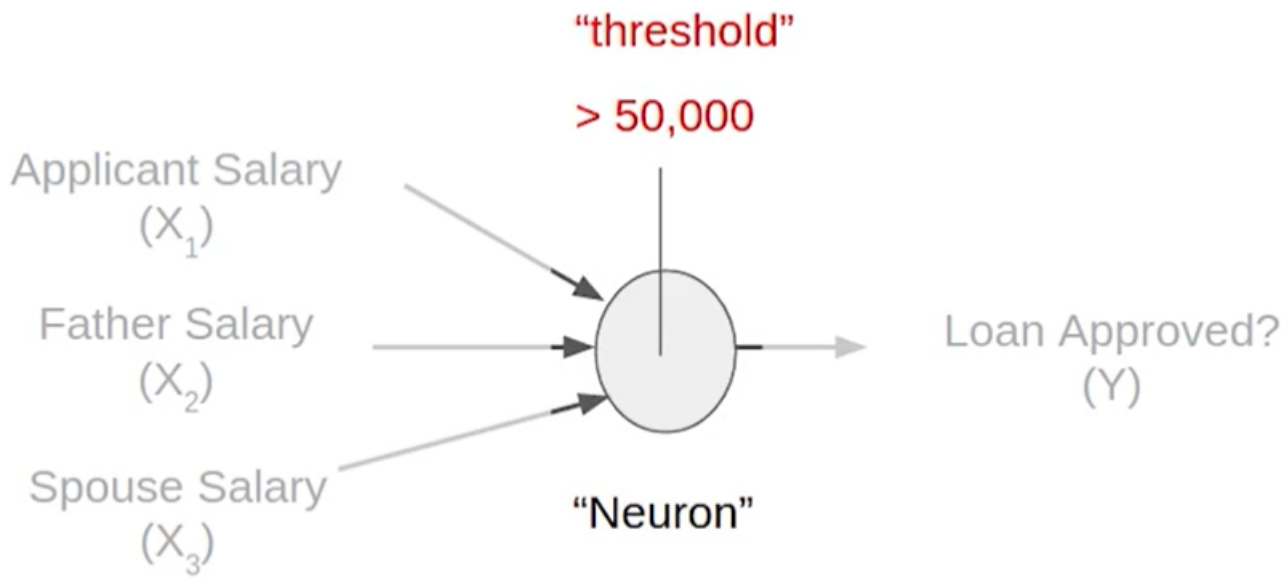
Deep Learning – Perceptron

More than one input variables can be considered.



Deep Learning – Perceptron

Consider that the addition of these input values is going to be compared with the threshold.



$$X_1 + X_2 + X_3 > T$$

$$X_1 + X_2 + X_3 - T > 0$$

$$X_1 + X_2 + X_3 + Bias > 0$$

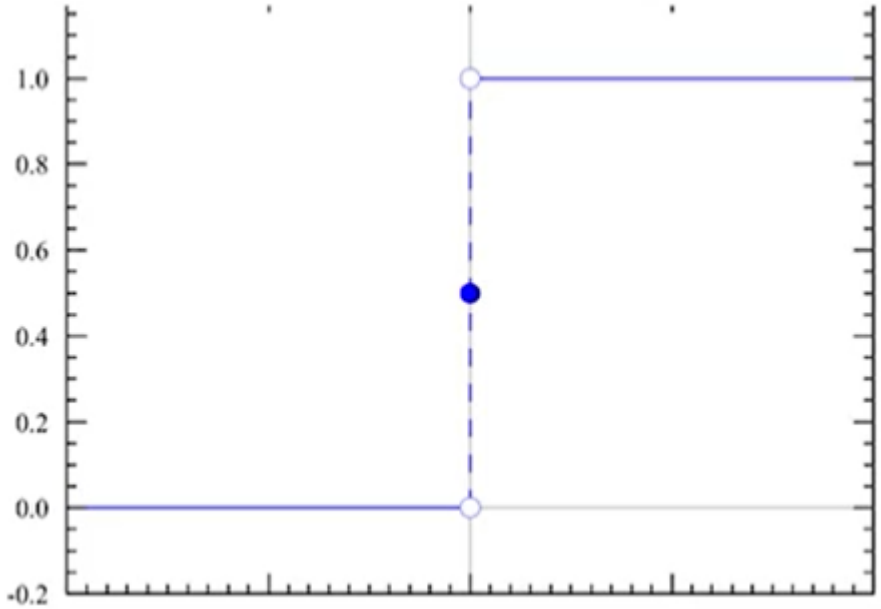
Total Income = Applicant Salary (X_1) + Father Salary (X_1) + Spouse Salary (X_3)

Total Income > threshold ?

$X_1 + X_2 + X_3 + Bias > 0$	1
$X_1 + X_2 + X_3 + Bias < 0$	0

Deep Learning – Perceptron

This follows the step function.

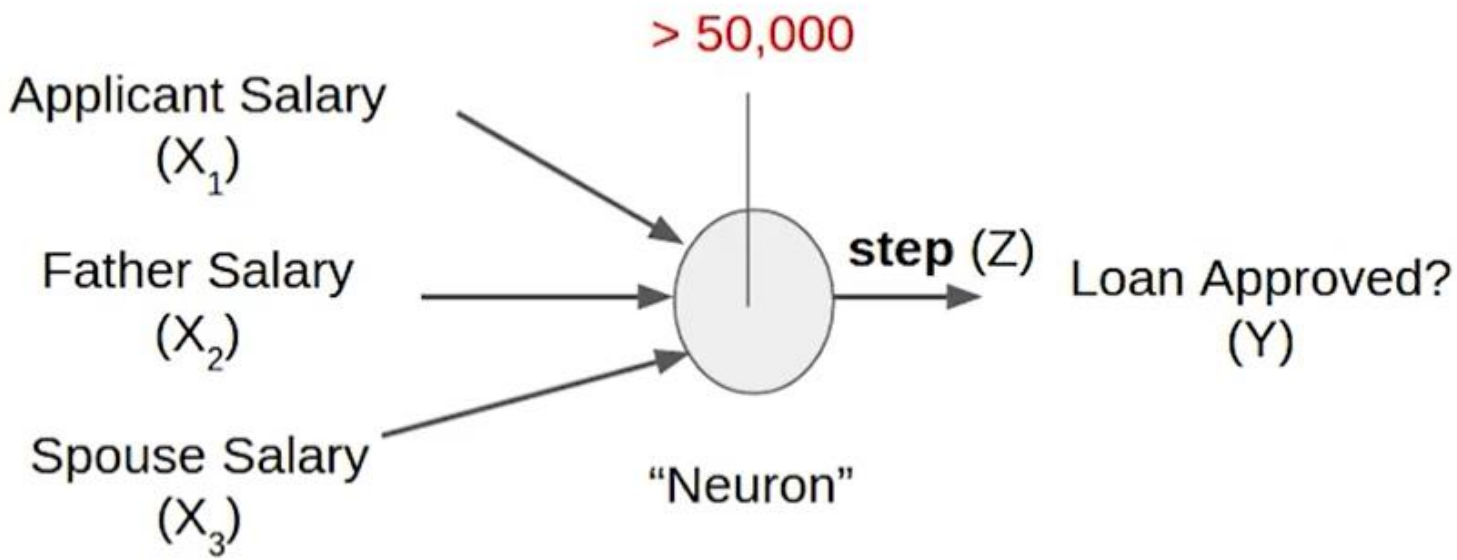


$$Z = X_1 + X_2 + X_3 + Bias$$

$$Step(Z) = Y = \begin{cases} 1 & Z > 0 \\ 0 & Z \leq 0 \end{cases}$$

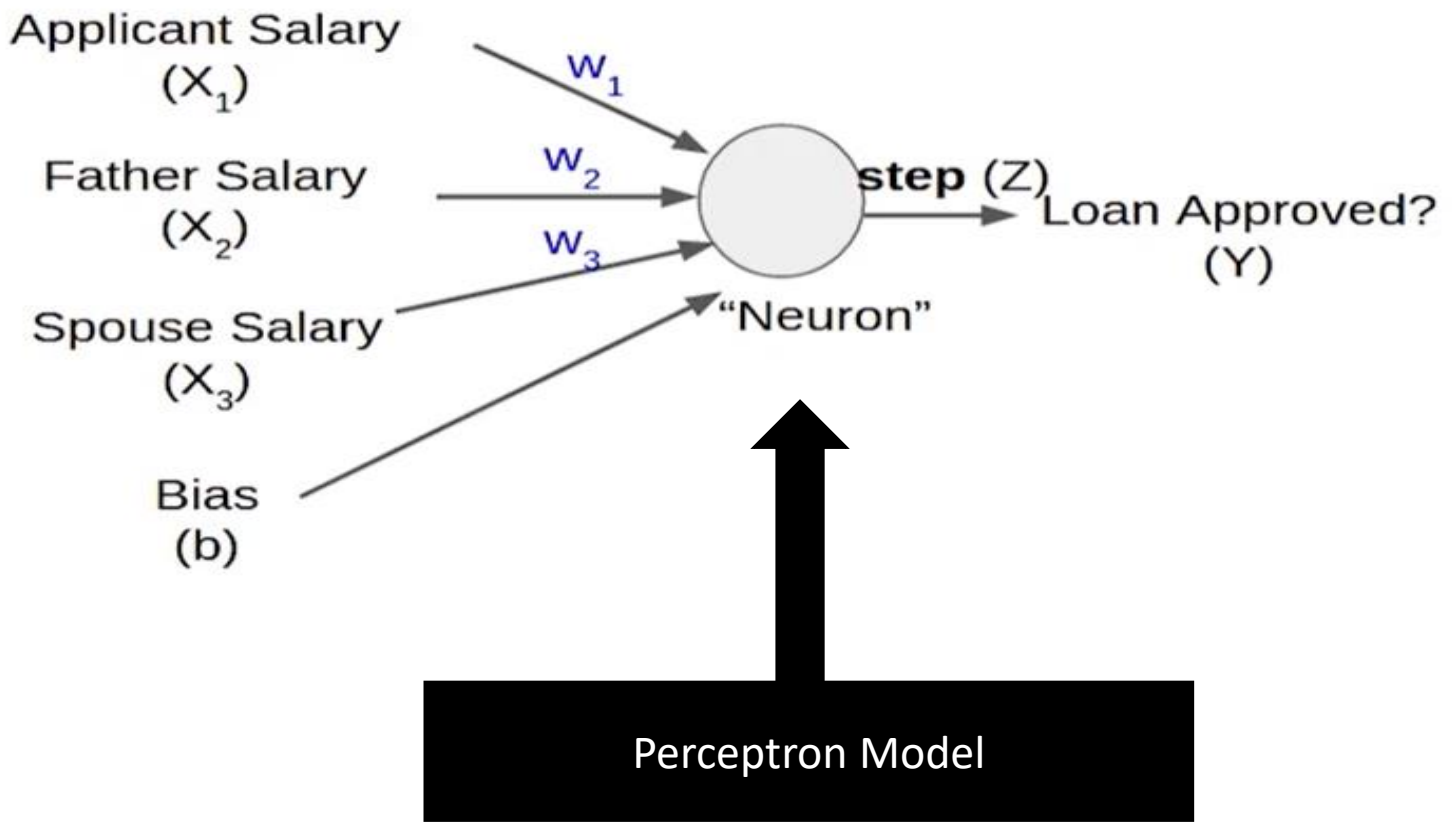
Deep Learning – Perceptron

Finally, the system can be obtained as,



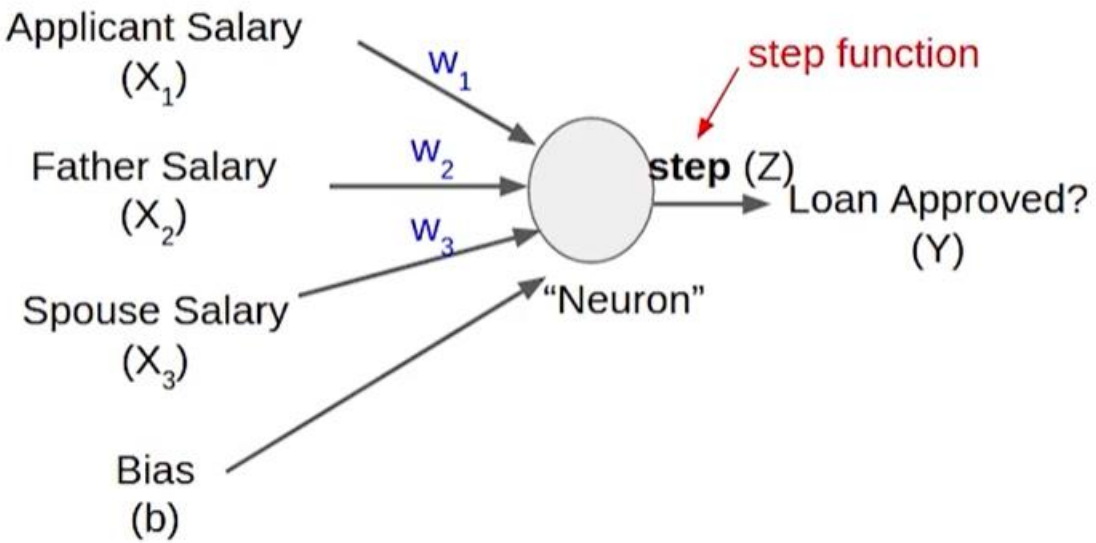
Deep Learning – Perceptron

The weight can be changed for each input.



Deep Learning – Perceptron

Weights and bias values can be changed and updated.



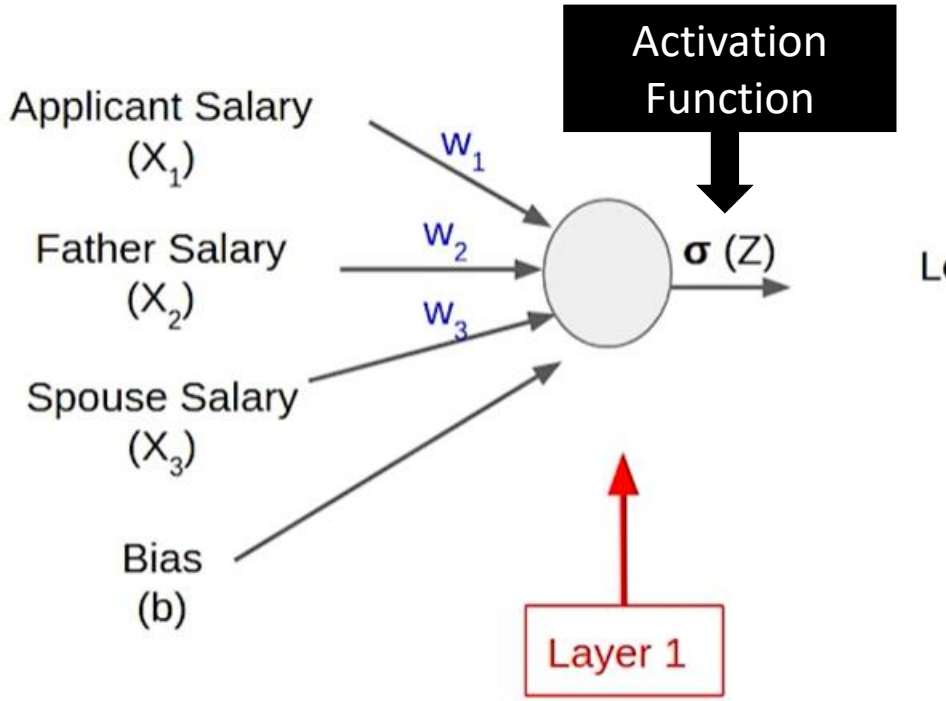
Sum of inputs = $X_1 * w_1 + X_2 * w_2 + X_3 * w_3$

$Z = X_1 * w_1 + X_2 * w_2 + X_3 * w_3 + b$ (bias)

$\hat{Y}$ (output) = **step** (Z)

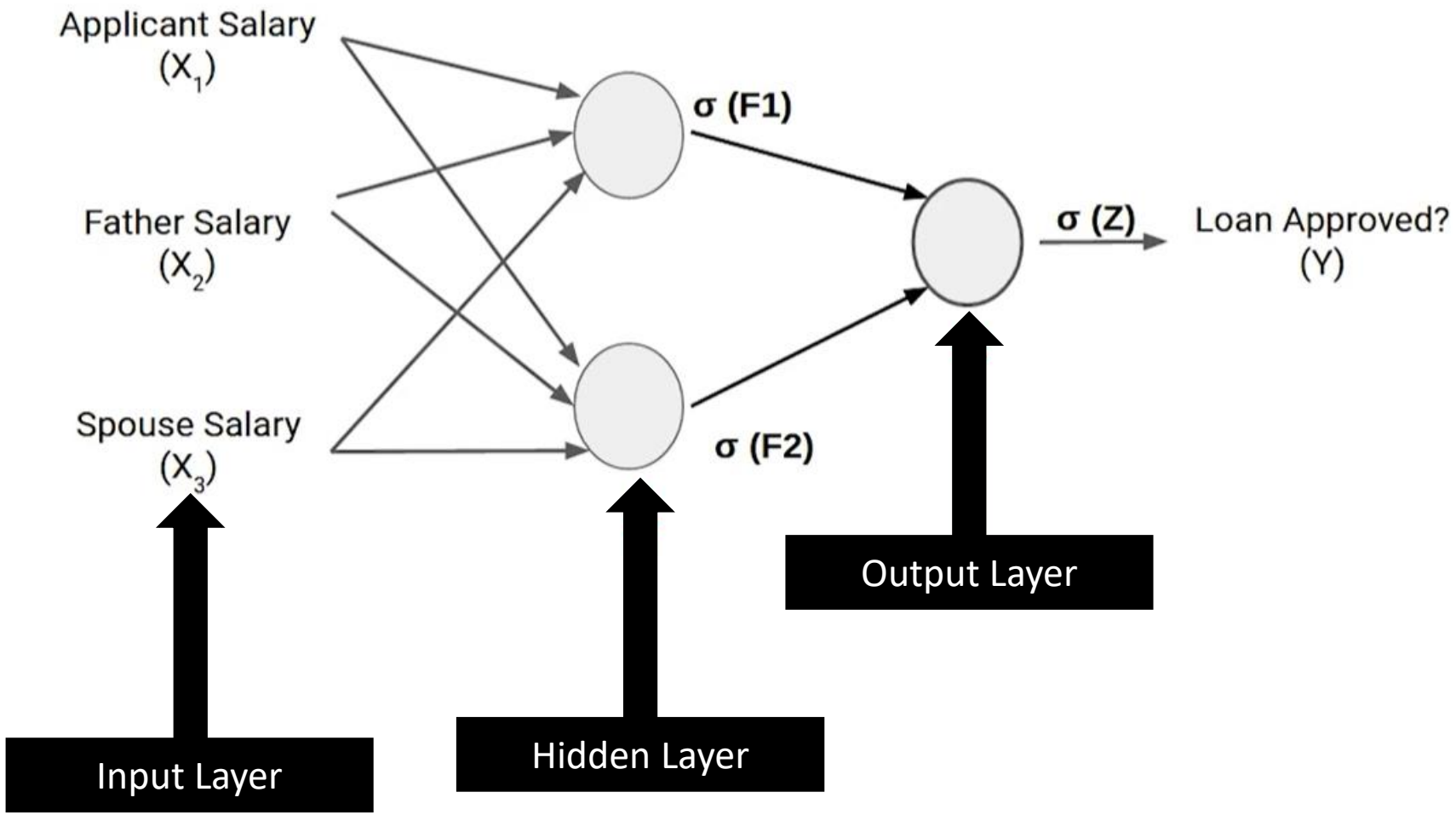
Deep Learning – Single Layer Perceptron

Here we have only one neuron in the hidden layer.



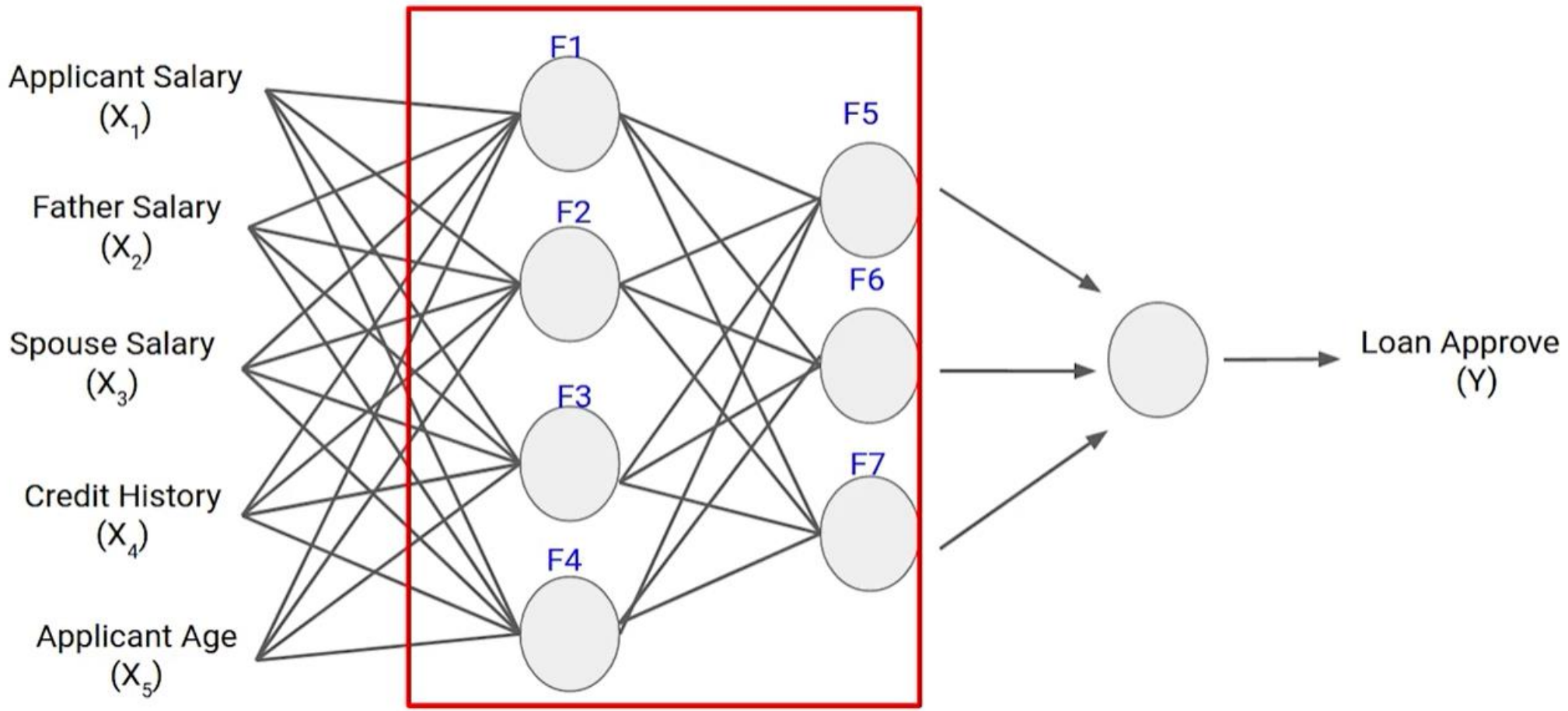
Deep Learning – Multi Layer Perceptron

We can have more than one neuron in the hidden layer.



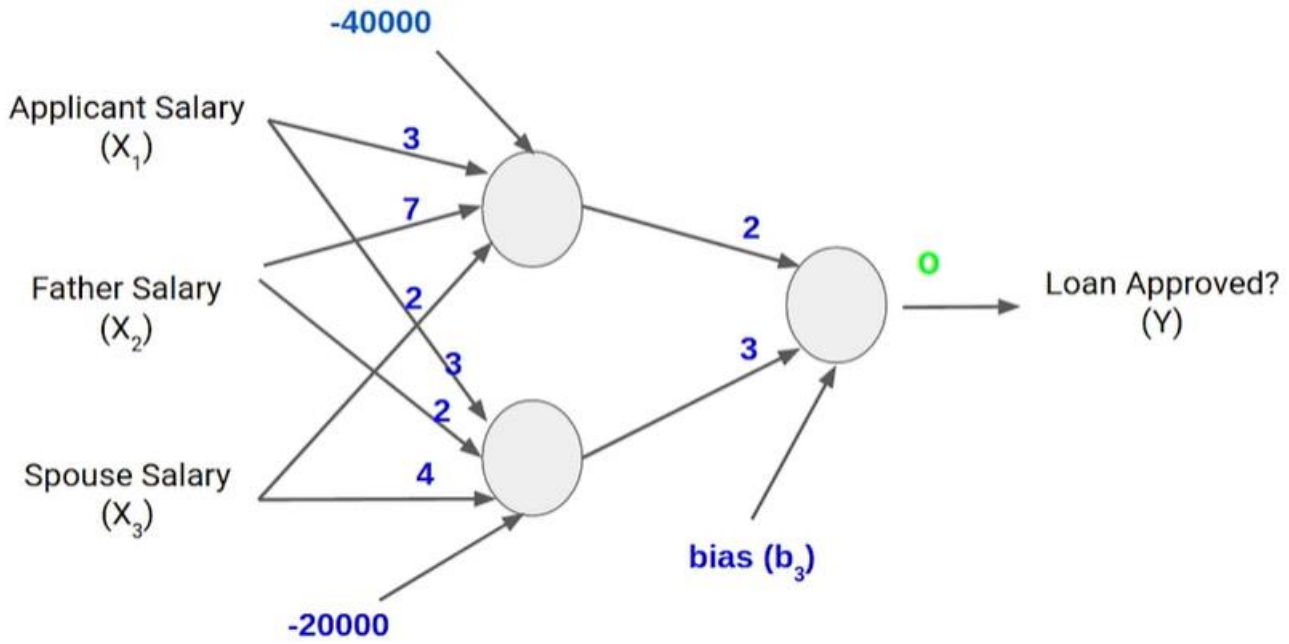
Deep Learning – Multi Layer Perceptron

More than one layers can be added.



Deep Learning – Forward & Backward Propagation

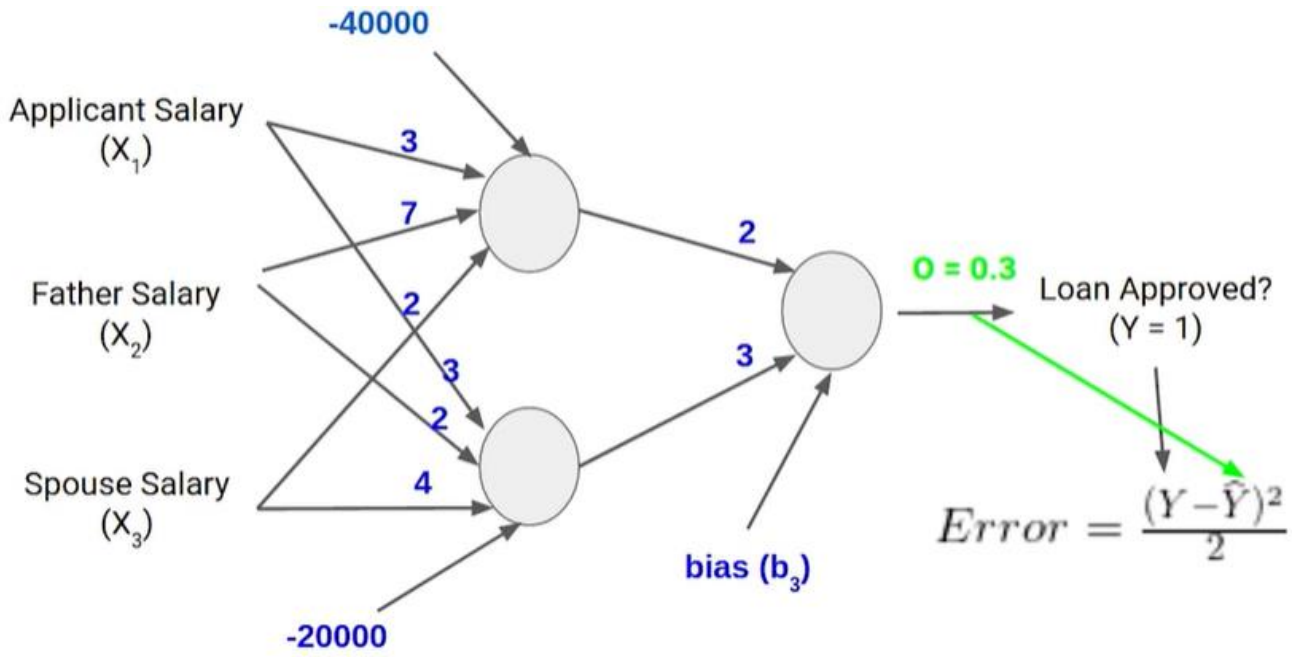
Consider the following example.



With these weights and bias values the output is obtained. This process is called the forward propagation.

Deep Learning – Forward & Backward Propagation

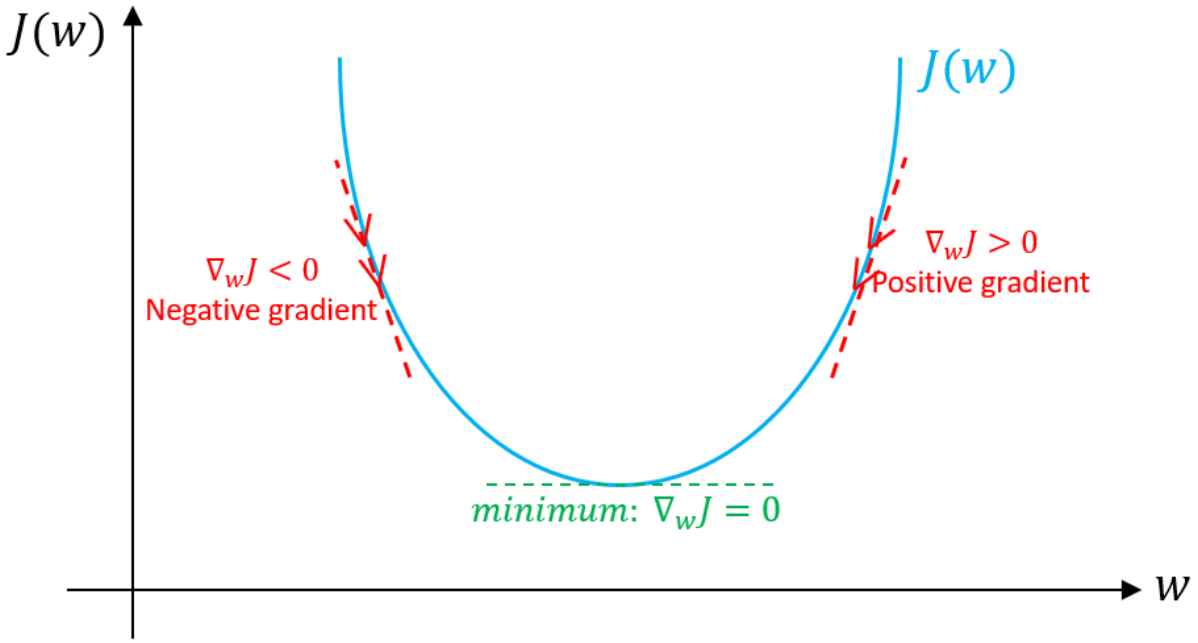
Consider the following example.



Weights and bias will be updated such that the error is minimized. This process is called the backward propagation.

Deep Learning – Gradient Descent Algorithm

This is the algorithm which is used for minimizing the errors by adjusting the weights and bias values.



Learning Rate

$$\theta_j = \theta_j - \alpha \frac{\partial}{\partial \theta_j} J(\theta)$$

Now,

$$\frac{\partial}{\partial \theta} J_\theta = \frac{\partial}{\partial \theta} \frac{1}{2m} \sum_{i=1}^{m} \left[h_\theta(x_i) - y_i \right]^2$$

$$\frac{\partial}{\partial \theta} J_\theta = \frac{1}{m} \sum_{i=1}^{m} \left(h_\theta(x_i) - y_i \right) \cdot \frac{\partial}{\partial \theta_j} \left(\theta x_i - y_i \right)$$

$$\frac{\partial}{\partial \theta} J_\theta = \frac{1}{m} \sum_{i=1}^{m} \left[\left(h_\theta(x_i) - y \right) x_i \right]$$

Therefore,

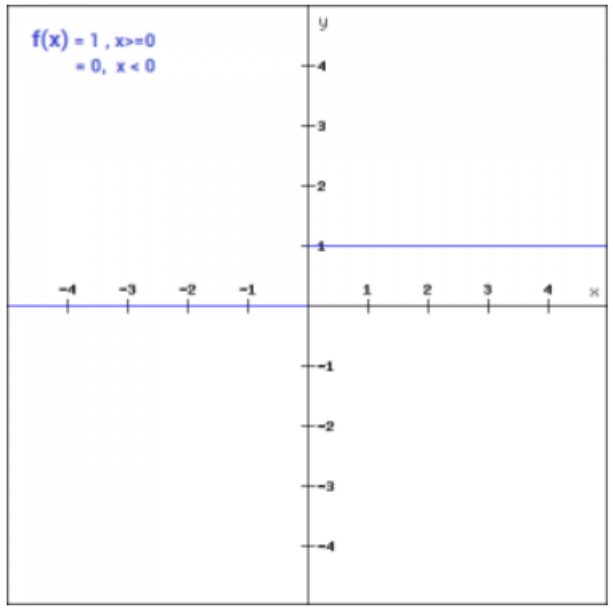
$$\theta_j := \theta_j - \frac{\alpha}{m} \sum_{i=1}^{m} \left[\left(h_\theta(x_i) - y_i \right) x_i \right]$$

This θ_j can be the weight or bias

This updating process stops when the error is not further minimized or when the number of iterations is completed.

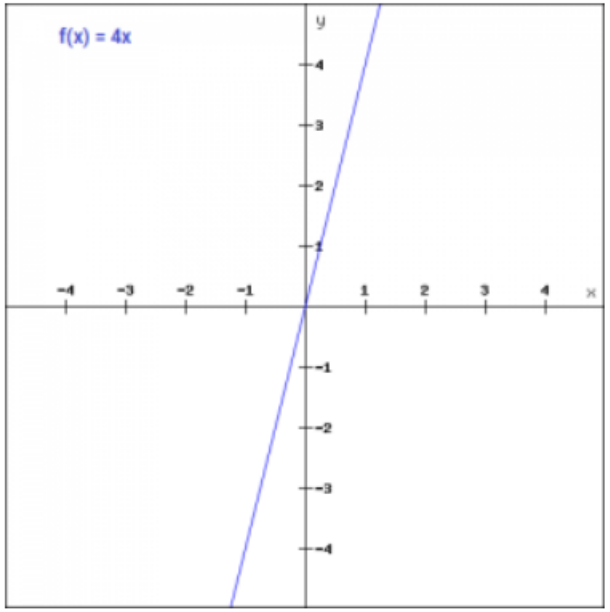
Deep Learning – Binary stepwise function

If the input to the activation function is greater than a threshold, then the neuron is activated, else it is deactivated,



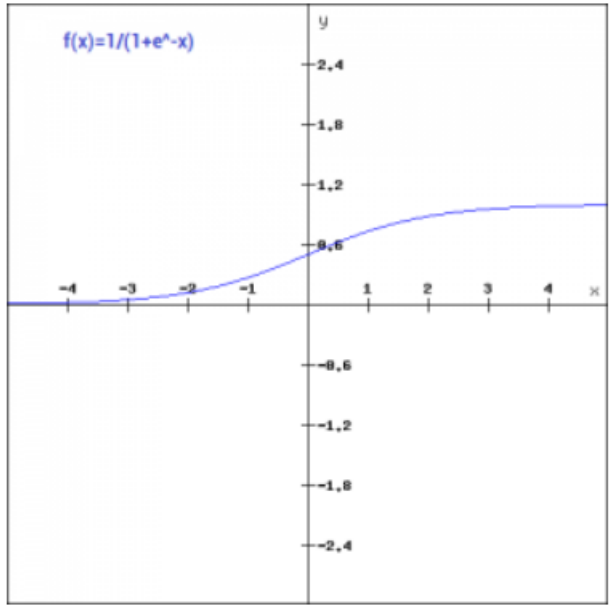
Deep Learning – Linear activation function

The activation is proportional to the input. These activation functions are mostly used in regression problems.



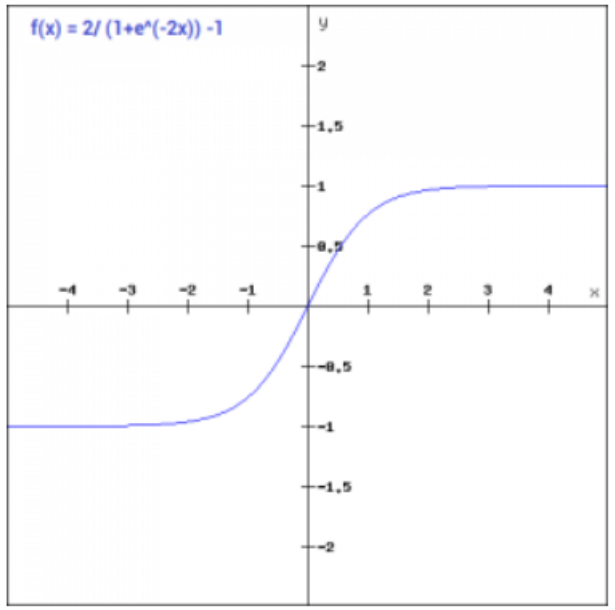
Deep Learning – Sigmoid function

It is one of the most widely used non-linear activation function. Sigmoid transforms the values between the range 0 and 1. Usually this is used for binary classifications.



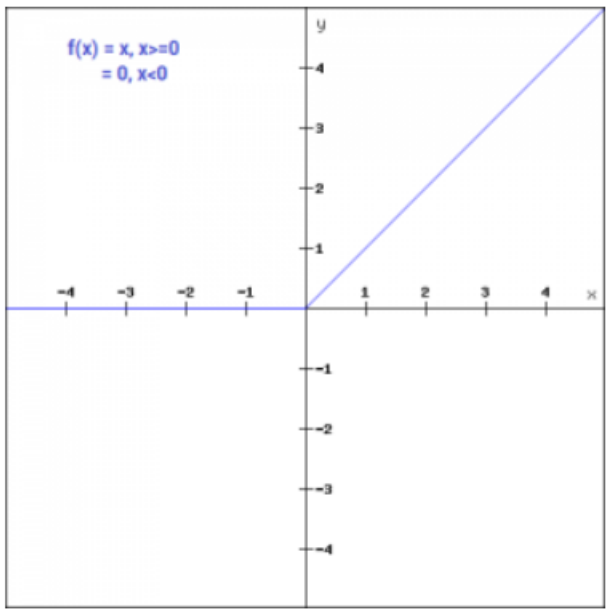
Deep Learning – Tanh function

The tanh function is very similar to the sigmoid function. The only difference is that it is symmetric around the origin. The range of values in this case is from -1 to 1. Thus the inputs to the next layers will not always be of the same sign.



Deep Learning – ReLU (Rectified Linear Unit) function

The ReLU function is another non-linear activation function that has gained popularity in the deep learning domain. ReLU stands for Rectified Linear Unit. The main advantage of using the ReLU function over other activation functions is that it does not activate all the neurons at the same time.



This means that the neurons will only be deactivated if the output of the linear transformation is less than 0. The plot below will help you understand this better-

Deep Learning – Softmax function

Softmax function is often described as a combination of multiple sigmoids. We know that sigmoid returns values between 0 and 1, which can be treated as probabilities of a data point belonging to a particular class. Thus sigmoid is widely used for binary classification problems.

The softmax function can be used for multiclass classification problems. This function returns the probability for a datapoint belonging to each individual class.

$$\sigma(\mathbf{z})_j = \frac{e^{z_j}}{\sum_{k=1}^{K} e^{z_k}} \quad \text{for } j = 1, \ldots, K.$$

Deep Learning – Loss Function

A loss function is used to optimize the parameter values in a neural network model. Loss functions map a set of parameter values for the network onto a scalar value that indicates how well those parameter accomplish the task the network is intended to do.

Regression Problem:

Loss Function: Mean Squared Error (MSE).

Binary Classification Problem:

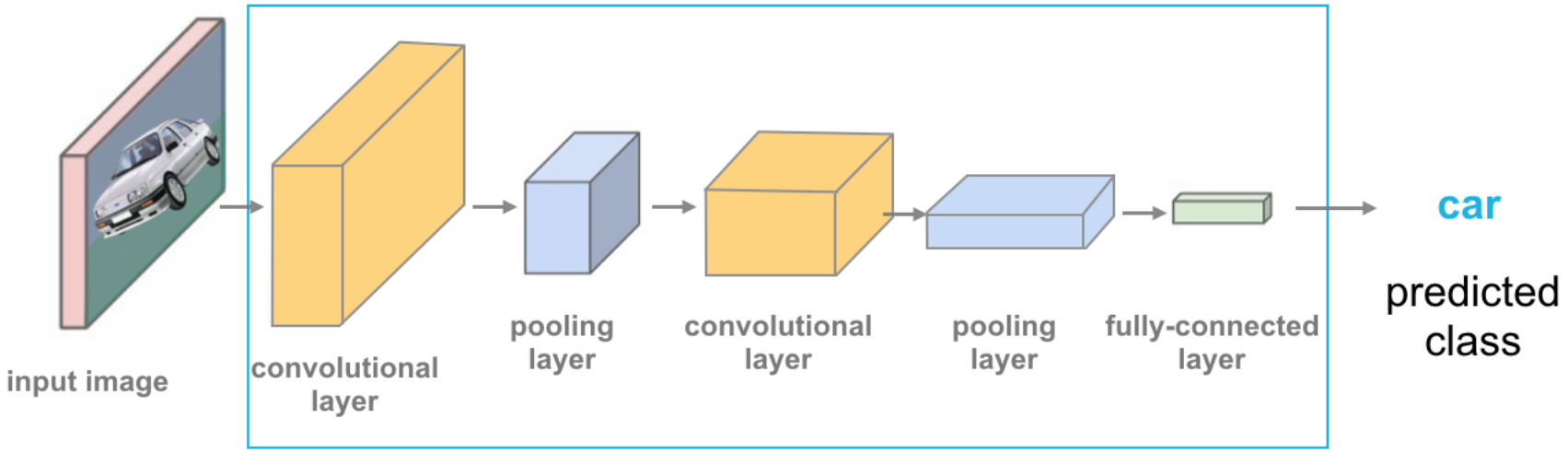
Loss Function: Cross-Entropy, also referred to as Logarithmic loss.

Multi-Class Classification Problem:

Loss Function: Cross-Entropy, also referred to as Logarithmic loss.

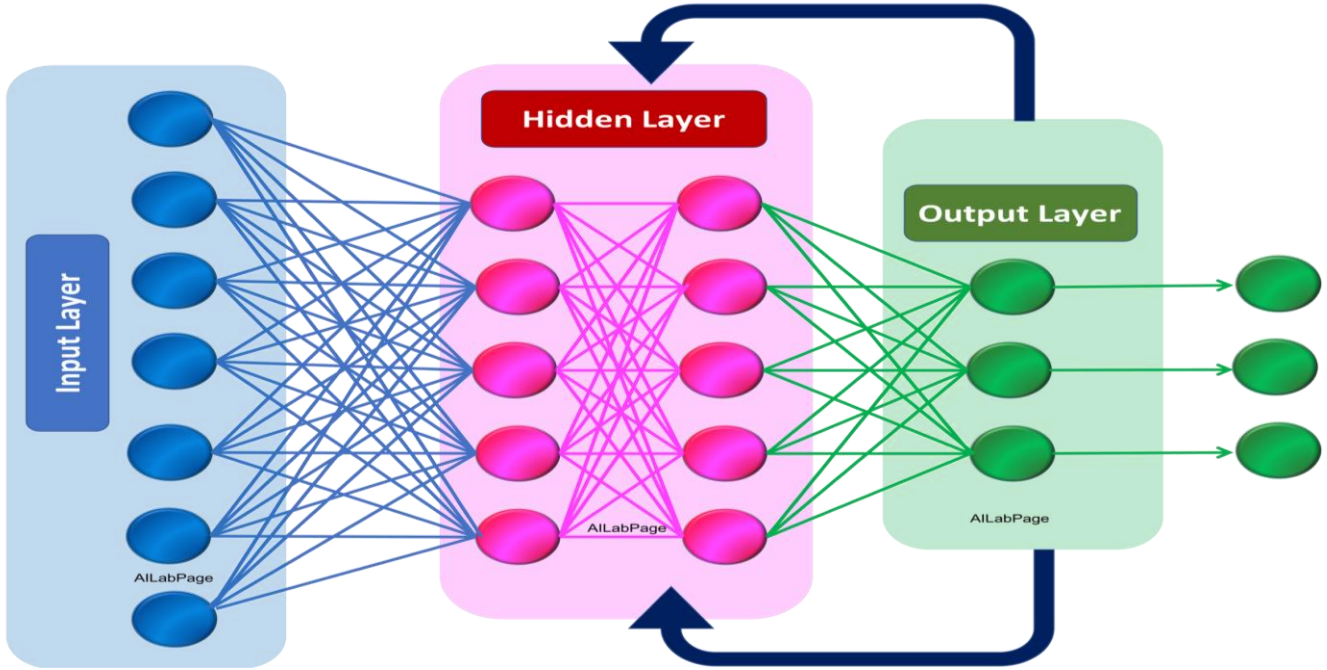
Deep Learning – Convolutional Neural Networks (CNN)

In deep learning, a convolutional neural network is a class of deep neural networks, most commonly applied to analyzing visual imagery. They are also known as shift invariant or space invariant artificial neural networks, based on their shared-weights architecture and translation invariance characteristics.



Deep Learning – Recurrent Neural Network (RNN)

A recurrent neural network (RNN) is a class of artificial neural networks where connections between nodes form a directed graph along a temporal sequence.



Machine Learning Libraries in Python