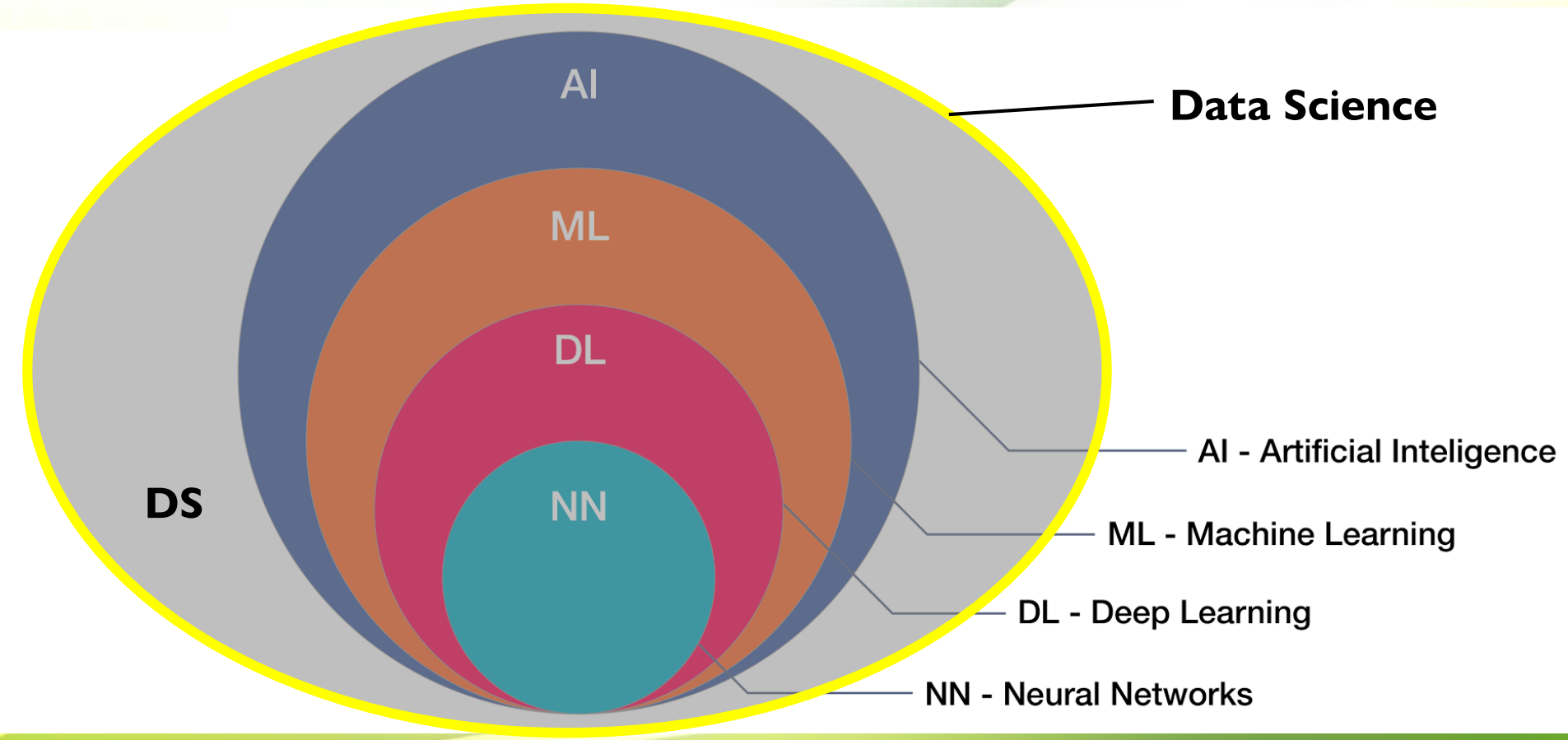
INTRODUCTION TO PREDICTIVE ANALYTICS

2nd Sem, MCA

CONTENT

- Introduction Predictive analytics
  - Predictive modeling
  - Model performance measure
  - Regression Models – Linear, Logistic and others
  - K-nearest neighbor
  - Classification & Regression model

PREDICTIVE ANALYTICS



DATA SCIENCE

- **Data is EVERYTHING;** a new form of revenue.
- Data gives better business insights; helps to uncover *(hidden)* patterns in data.
- Example:
  - One guy order for a computer. He also purchases a mouse and keyboard.
  - Data modeling will build this pattern.
  - Companies use this patter/relation for better business policy. Predictive analytics for future buying.
  - Companies use DS to build recommendation engines. Prescriptive analytics in DS.
- Various algorithms could be applied on data to get more accurate results.
- Running these algorithms on huge datasets needs AI, ML, DL.
- ML is used in DS to make predictions by discovering hidden patterns in data.

ARTIFICIAL INTELLIGENCE

- AI enables Machine to think.
- Adding intelligence to our system in artificial way.
- Ability that enables machines to understand data, learn from data, and make decisions based on patterns hidden in the data, or inferences that could otherwise be very difficult *(almost impossible)* for humans to make manually.
- AI enables machines to adjust/use the gained "knowledge" based on new inputs that were not part of the data used for training these machines.
- Collection of mathematical algorithms that make computers understand relationships between different types & pieces of data to use knowledge to make decisions that could be accurate to a very high degree.

MACHINE LEARNING

- Machine's ability to learn.
- ML is an implementation of AI.
- Establish Relationship between independent & dependent variables present in data.
- ML is used in situations where machine should learn from huge amounts of data given to it ***(training dataset)***, and then apply that knowledge on new pieces of data that streams into the system.
- After learning/training phase is complete (with training data); ML model is tested on data which machine never encountered before ***(testing dataset).***
- Statistical tool to analyze data to get conclusive knowledge.

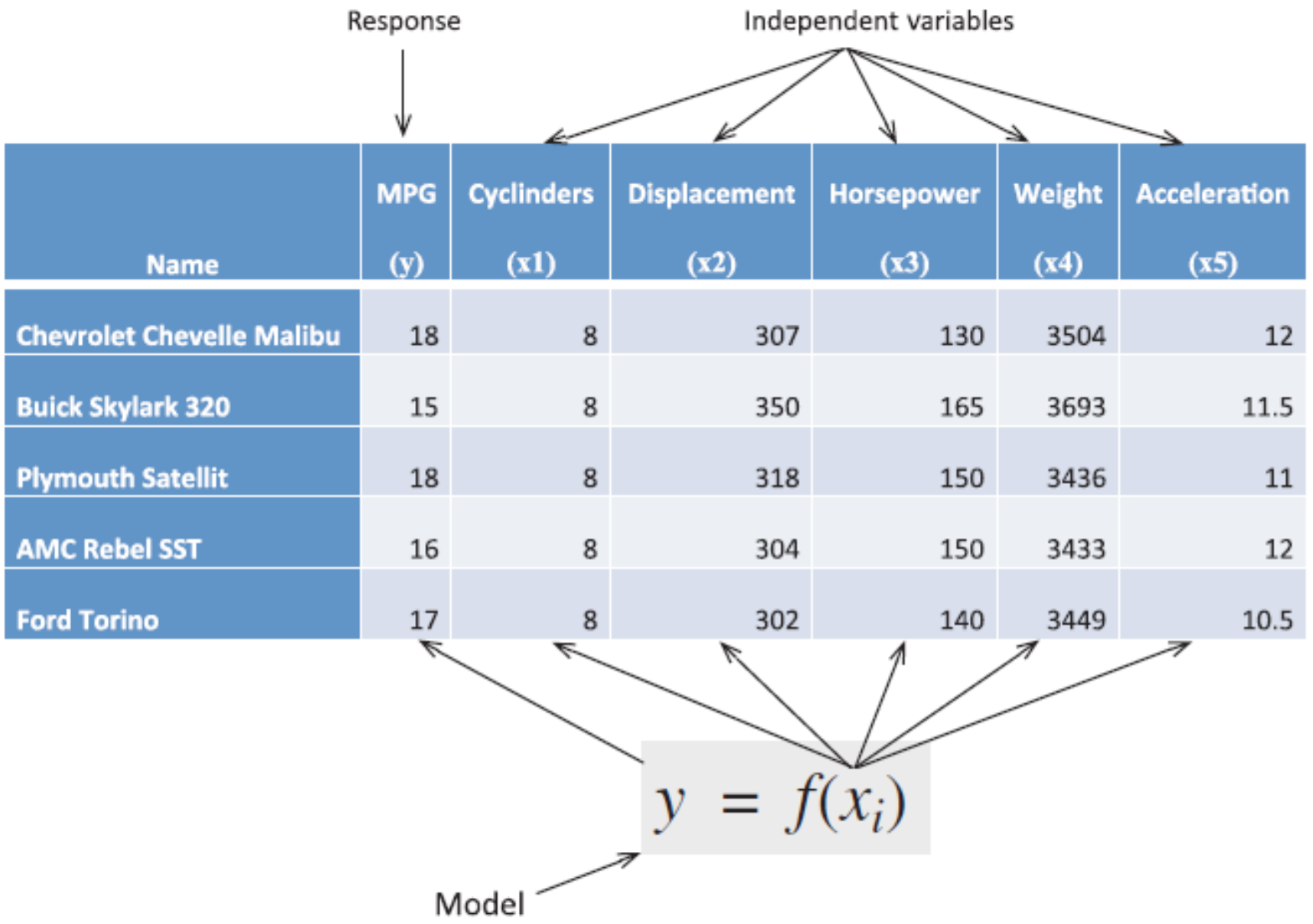
DEEP LEARNING

- Fills the gaps of ML limitations.
- ML involves structured data.
- DL also deals with unstructured data (video, audio, text, social media posts and images — anything and everything that humans communicate with that are not numbers or metric reads).
- Training dataset is relatively small for ML.
- When huge amounts of data to train a model, with data having too many features, and if high level of accuracy is critical, DL is more appropriate.
- DL requires much powerful hardware to run on (GPUs).
- Takes significantly more time to train models; generally more difficult to implement compared to ML.

Predictive Modeling

- **Predictive modeling** is a commonly used statistical technique to predict future behavior.
- Technology that analyzes historical/current data and generate model to help predict future outcomes.
- Data is collected, statistical model is formulated, predictions are made, and model is validated (or revised) as additional data becomes available.
- Various algorithms in ML used for *prediction problems, classification problems, regression problems,* etc.
- Predictive analytics models are:
  - *Classification model*
  - *Clustering model*
  - *Forecast model*
  - *Outliers model*
  - *Time series model*

Predictive Modeling

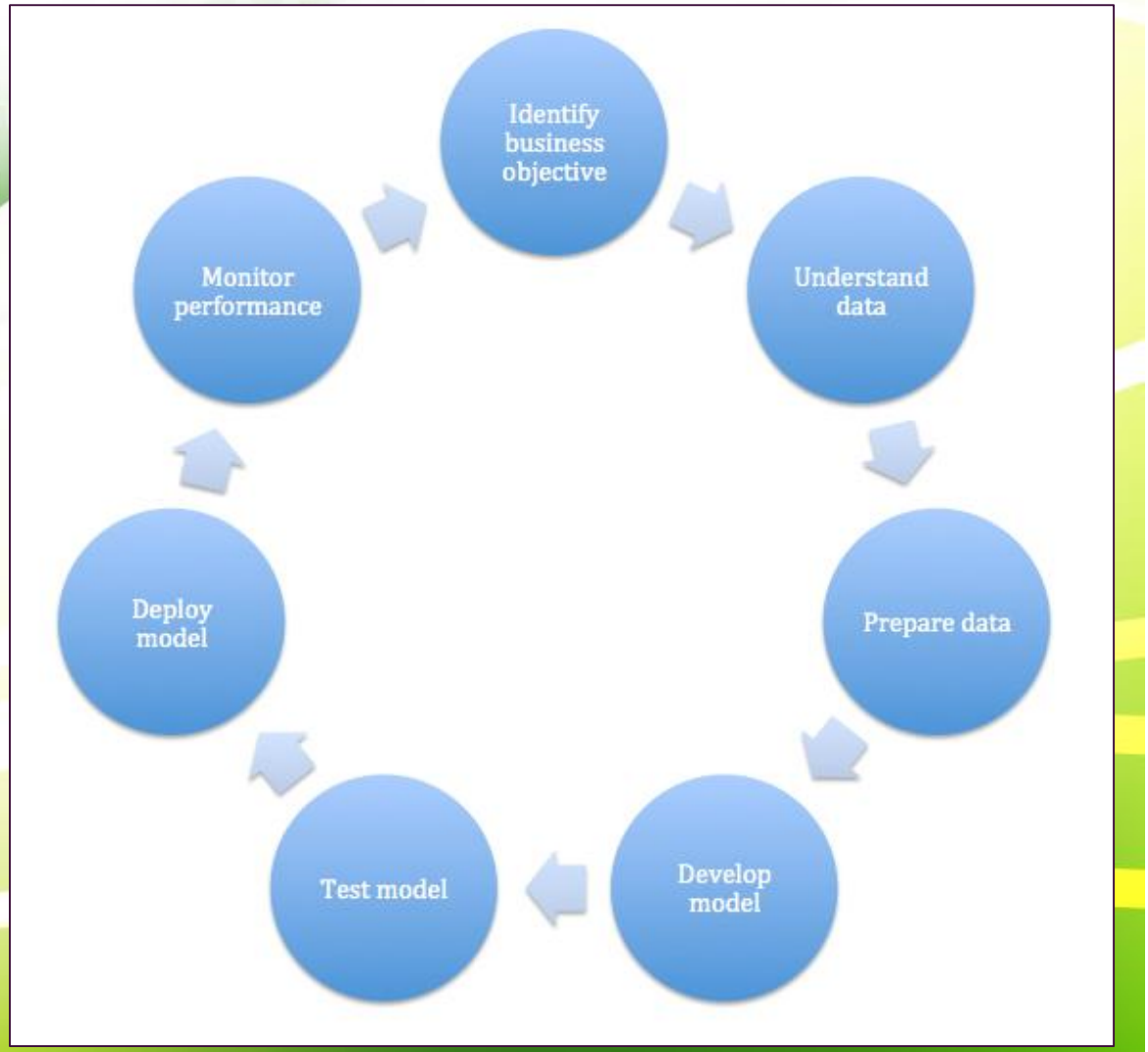


Name	MPG (y)	Cyclinders (x1)	Displacement (x2)	Horsepower (x3)	Weight (x4)	Acceleration (x5)
Chevrolet Chevelle Malibu	18	8	307	130	3504	12
Buick Skylark 320	15	8	350	165	3693	11.5
Plymouth Satellit	18	8	318	150	3436	11
AMC Rebel SST	16	8	304	150	3433	12
Ford Torino	17	8	302	140	3449	10.5

Predictive Modeling

Phases of Predictive Modeling

- *Understand Business Objective*
- *Define Modelling Goals*
- *Select/Collect Data*
- *Prepare Data*
- *Analyse and Transform Variables,* ***Sampling***
- ***Model*** *Selection, Develop Models (Training)*
- *Validate Models (Testing), Optimize, Profitability*
- *Document Methodology and Models*
- *Implement Models*
- *Monitoring and Performance Tracking*



Predictive Modeling

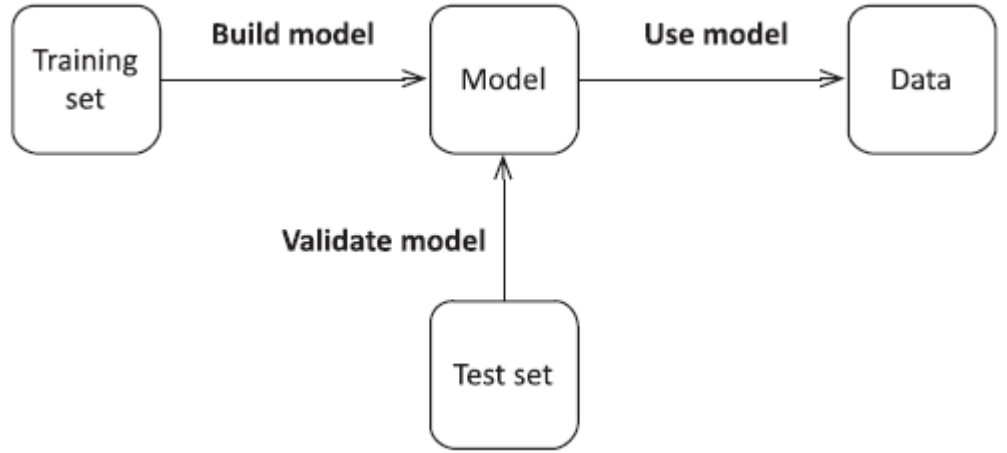
Predictive modeling limitations :

- Errors in data labeling.
- Shortage of massive data sets needed to train machine learning.
- Machine's inability to explain what and why it did what it did.
- Generalizability of learning, or rather lack thereof.
- Bias in data and algorithms.

Predictive Modelling

- ***Training** set:* data set used to build a model.
- ***Test set**:* data set used to test the model
- **Cross-validation:** validating model efficiency by training it on subset of input data and testing on previously unseen subset of input data.
- Three steps involved in cross-validation:
  - Reserve some portion of sample data-set.
  - Using the rest data-set train the model.
  - Test the model using the reserve portion of the data-set.
- Methods used for Cross-Validation:
  - Validation Set Approach (50%)
  - Leave-P-out cross-validation (p, n-p)
  - Leave one out cross-validation (1, n-1)
  - K-fold cross-validation
  - Stratified k-fold cross-validation

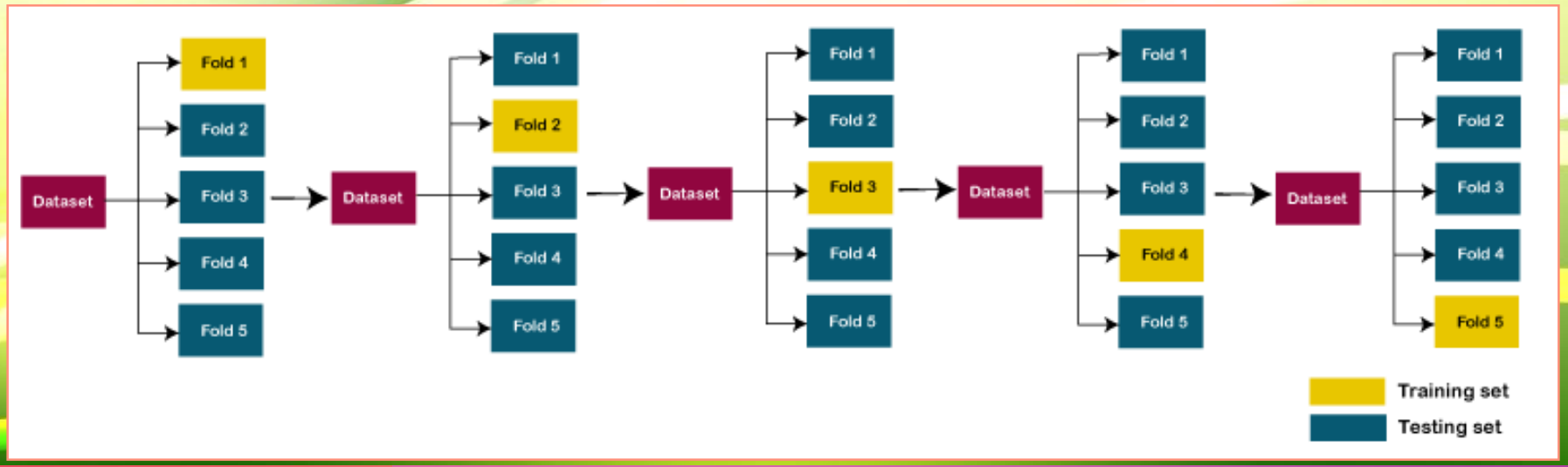
Name	MPG (y)	Cyclinders (x1)	Displacement (x2)	Horsepower (x3)	Weight (x4)	Acceleration (x5)
Chevrolet Chevelle Malibu	18	8	307	130	3504	12
Buick Skylark 320	15	8	350	165	3693	11.5
Plymouth Satellit	18	8	318	150	3436	11
AMC Rebel SST	16	8	304	150	3433	12
Ford Torino	17	8	302	140	3449	10.5



Predictive Modelling

K-fold Cross-validation:

1. Split input dataset into K groups/folds (equal size)
2. Repeat this step by each group on rotation:
   - Take one group as reserve or test dataset.
   - Use remaining groups as training dataset.
   - Fit the model on training set and evaluate performance of the model using the test set.
3. Accuracy of the model is based on average of the k scores.

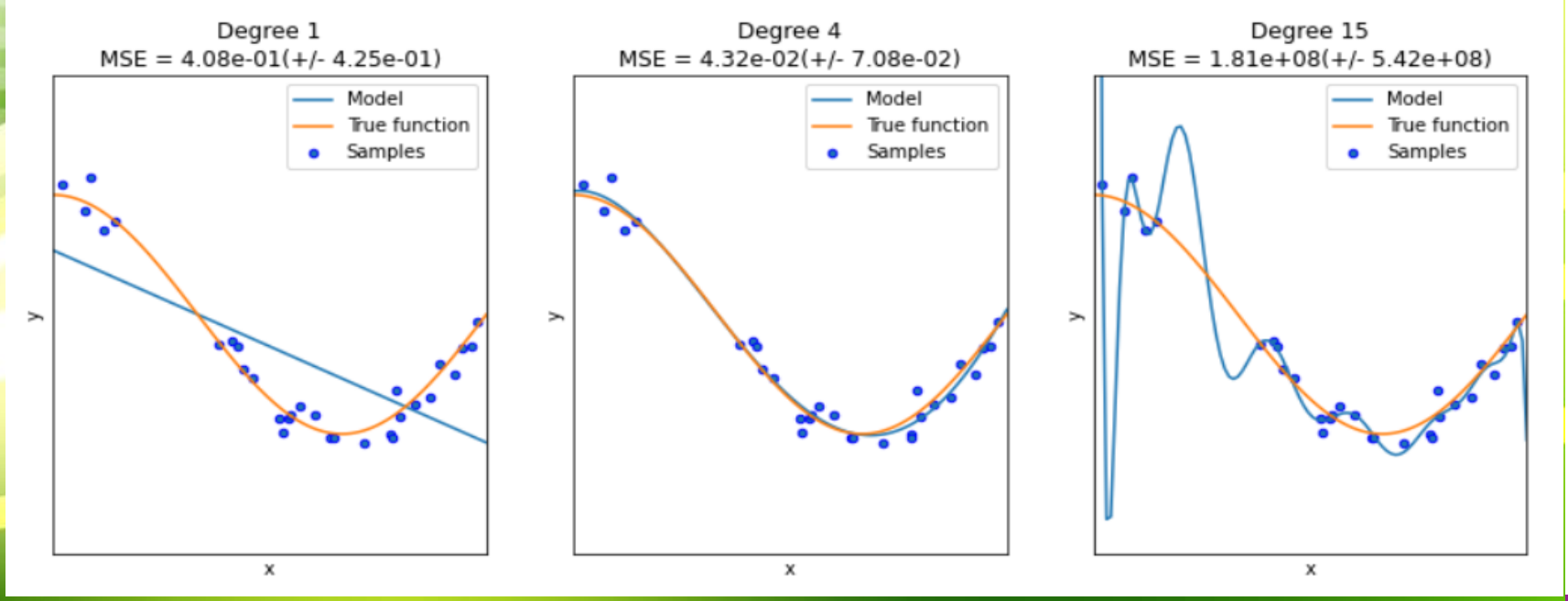


Predictive Modelling

- In statistics a **fit** is, **how close model is to target class/function/value**.
- Key components in ML modelling:
  - **Signal:** true underlying pattern of data that helps ML model to learn from data.
  - **Noise:** unnecessary and irrelevant data that reduces performance of model.
  - **Bias:** measure of model accuracy.
    - difference between predicted values and actual values.
    - prediction error that is introduced in model due to oversimplifying/optimizing ML algorithms.
  - **Variance:** If ML model performs well (low error) with training dataset, but does not perform well (high error) with test dataset.
- Overfitting and underfitting are two biggest causes of poor performance of ML models.

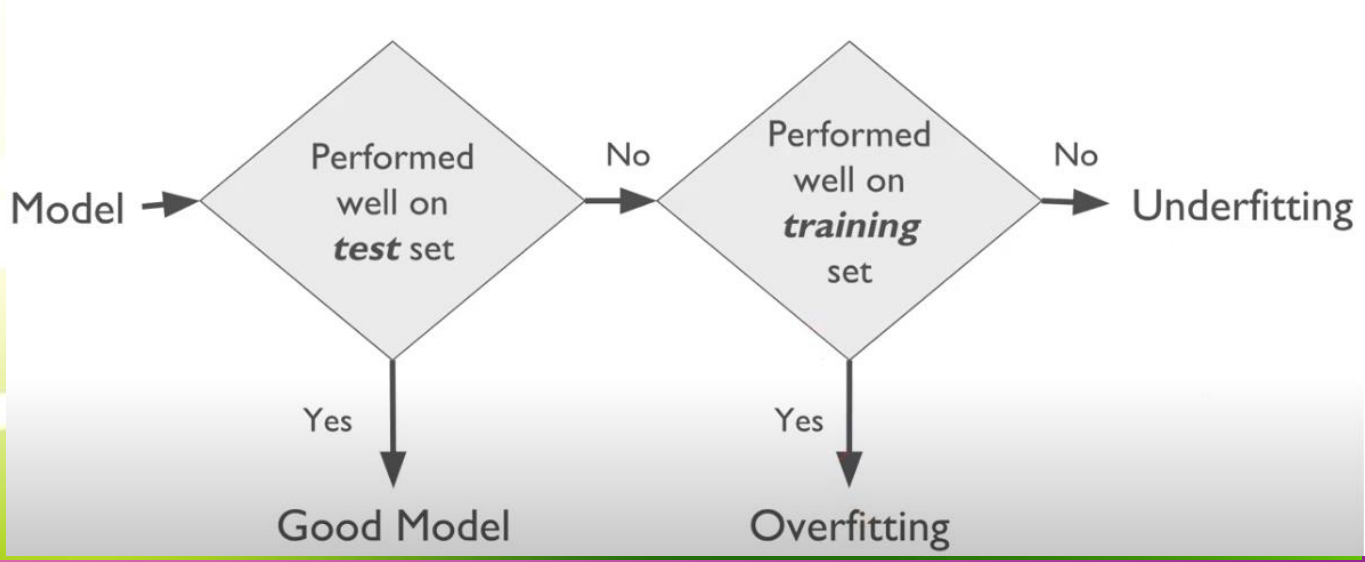
Predictive Modelling

- **Overfitting** model tries to cover all/more than required data points present in given dataset.
  - Model starts catching noise/inaccurate values present in dataset.
  - model can't generalize or fit well on unseen dataset, and reduce model accuracy.
  - Error on testing or validation dataset is much greater than error on training dataset.
  - Overfitted model has low bias and high variance.
- Avoiding Overfitting:
  - Cross-Validation
  - Training with more clean data
  - Removing features
  - Early stopping the training
  - Regularization
  - Ensembling



Predictive Modelling

- Opposite of overfitting is underfitting.
- **Underfitting** when model is not able to capture underlying trend of data.
  - To avoid overfitting in model, volume/time of training data is compromised.
  - Hence, model is not able to learn enough from training data, and hence it reduces accuracy and produces unreliable predictions.
  - Underfitted model has high bias and low variance.
- Avoid underfitting:
  - increasing training time/volume of model.
  - increasing number of features.



Predictive Modelling - Confusion Matrix

ACTUAL	Course-1	Course-2	Course-3
Pass	90	70	80
Fail	30	50	40

PREDICTED	Course-1	Course-2	Course-3
Pass	80	50	70
Fail	40	70	50

Course-1

	Predicted Pass	Predicted Fail	Total
Actual Pass	70	20	90
Actual Fail	10	20	30
Total	80	40	

Predictive Modelling - Confusion Matrix

Contingency Table for Predictive modelling Performance metrics.

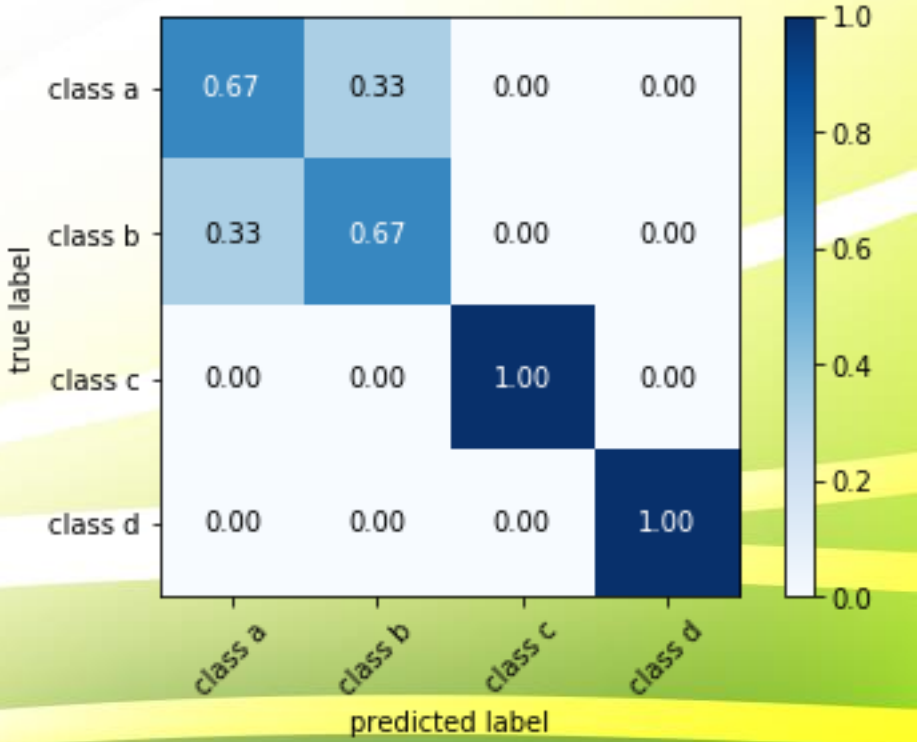
Confusion matrix: N X N matrix, where N is number of classes being predicted.

- **Error / residual:** difference between predicted value and actual value.
- **Accuracy / *Concordance***: proportion of total number of predictions that were correct.
- **Positive Predictive Value/Precision**: proportion of positive cases that were correctly identified.
- **Negative Predictive Value** : proportion of negative cases that were correctly identified.
- **Sensitivity or Recall** : proportion of actual positive cases which are correctly identified.
- **Specificity** : proportion of actual negative cases which are correctly identified.

Confusion Matrix		Target			
		Positive	Negative		
Model	Positive	a	b	*Positive Predictive Value*	a/(a+b)
	Negative	c	d	*Negative Predictive Value*	d/(c+d)
		Sensitivity	*Specificity*	**Accuracy** = (a+d)/(a+b+c+d)	
		a/(a+c)	d/(b+d)		

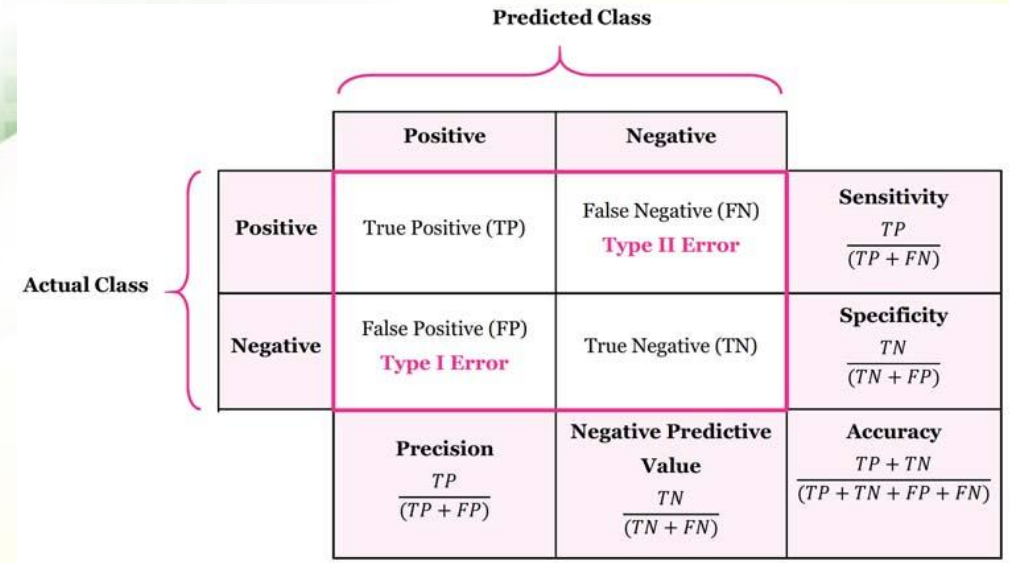
Predictive Modelling - Confusion Matrix

		Predicted Class: Positive	Predicted Class: Negative	
Actual Class	Positive	True Positive (TP)	False Negative (FN) **Type II Error**	**Sensitivity** $\frac{TP}{(TP+FN)}$
	Negative	False Positive (FP) **Type I Error**	True Negative (TN)	**Specificity** $\frac{TN}{(TN+FP)}$
		Precision $\frac{TP}{(TP+FP)}$	**Negative Predictive Value** $\frac{TN}{(TN+FN)}$	**Accuracy** $\frac{TP+TN}{(TP+TN+FP+FN)}$



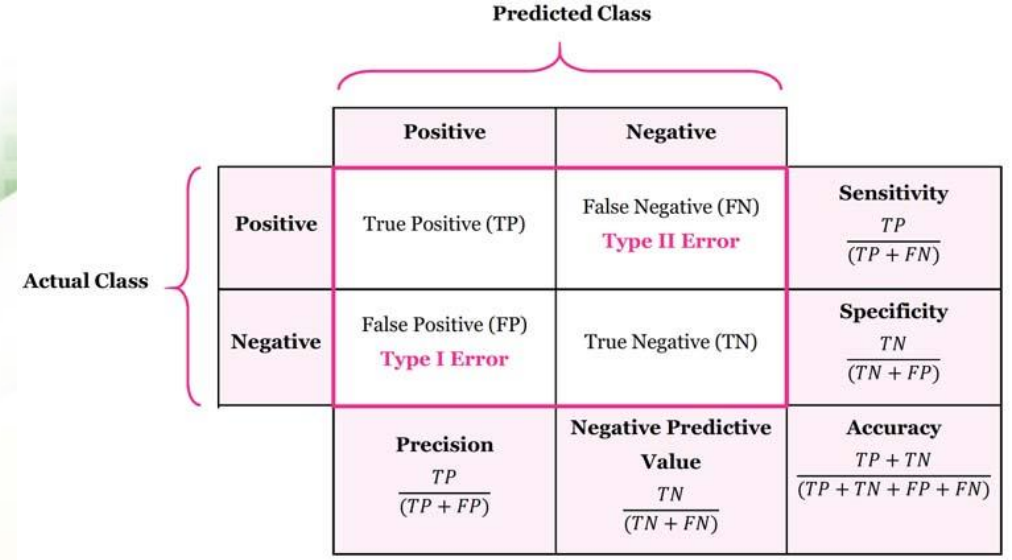
Predictive Modelling - Confusion Matrix

- **Classification accuracy** → total number of correct predictions divided by total number of predictions made for a dataset.
  - Accuracy is not always appropriate for all problems.



- Alternative to using classification accuracy → precision and recall metrics.
  - **Precision** → number of positive class predictions that actually belong to the positive class.
  - **Recall (sensitivity)** → number of positive class predictions made out of all positive examples in the dataset.
  - **F-Measure** → A single score that balances both the concerns of precision and recall.

Predictive Modelling - Confusion Matrix



- **Precision** → number of correct positive predictions made.

  Precision = TruePositives / (TruePositives + FalsePositives)

- 0.0 for no precision and 1.0 for full or perfect precision.

- A model makes predictions and predicts 120 examples as belonging to the positive class, 90 of which are correct, and 30 of which are incorrect.

  Precision = TruePositives / (TruePositives + FalsePositives) = 90 / (90 + 30) = 90 / 120 = 0.75

- Same dataset, another model predicts 50 examples belonging to the positive class, 45 of which are true positives and five of which are false positives.

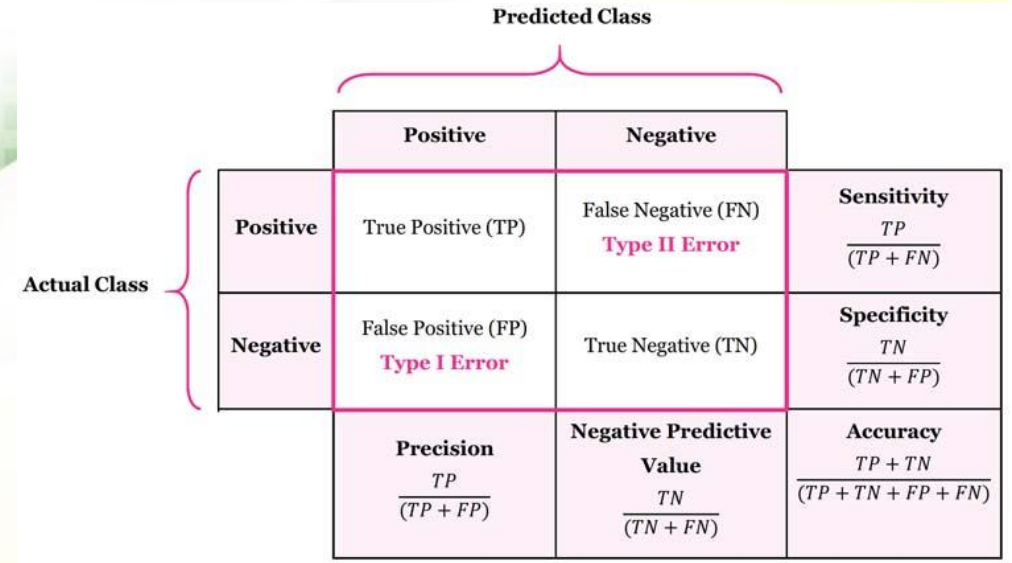
  Precision = TruePositives / (TruePositives + FalsePositives) = 45 / (45 + 5) = 45 / 50 = 0.90

Predictive Modelling - Confusion Matrix

- **Recall** → number of correct positive predictions made out of all positive predictions that could have been made.

  Recall = TruePositives / (TruePositives + FalseNegatives)

- 0.0 for no recall and 1.0 for full or perfect recall.



- A model makes predictions and predicts 90 of the positive class predictions correctly and 10 incorrectly.

  Recall = TruePositives / (TruePositives + FalseNegatives)

  Recall = 90 / (90 + 10) = 90 / 100 = 0.9

Predictive Modelling - Confusion Matrix

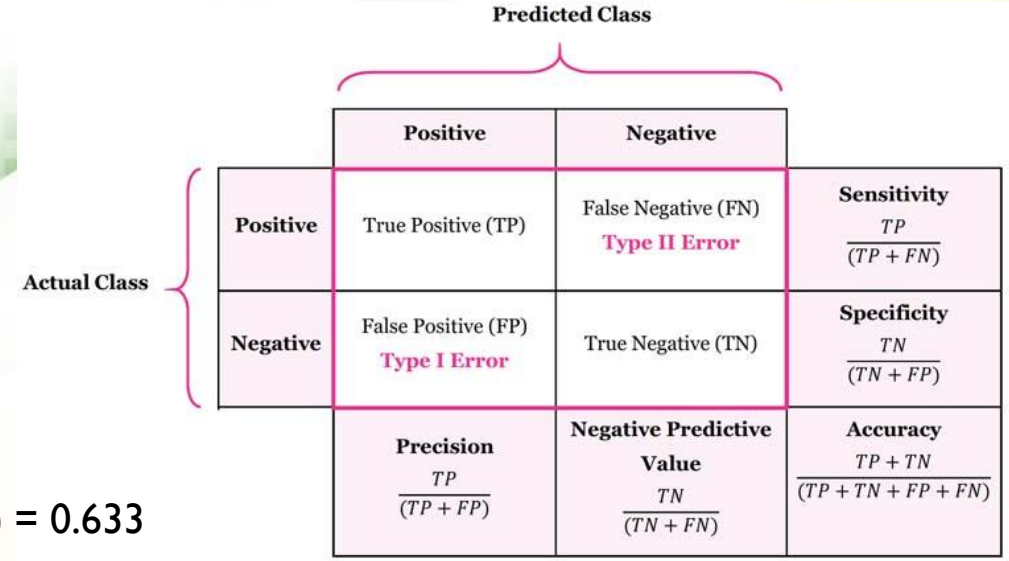
		Predicted Class: Positive	Predicted Class: Negative	
Actual Class	Positive	True Positive (TP)	False Negative (FN) **Type II Error**	**Sensitivity** $\frac{TP}{(TP+FN)}$
	Negative	False Positive (FP) **Type I Error**	True Negative (TN)	**Specificity** $\frac{TN}{(TN+FP)}$
		Precision $\frac{TP}{(TP+FP)}$	**Negative Predictive Value** $\frac{TN}{(TN+FN)}$	**Accuracy** $\frac{TP+TN}{(TP+TN+FP+FN)}$

- Maximizing precision will minimize the number false positives.
- Maximizing the recall will minimize the number of false negatives.
  - For excellent predictions both high precision and high recall needed.
  - Neither precision or recall tells whole story.
    - *Excellent precision with terrible recall, or terrible precision with excellent recall!!!*
  - Increases in recall often come at the expense of decreases in precision, and vice-versa.
- Instead of picking any one measure, a new metric can be used that combines both precision and recall into one score → F/F1-score (harmonic mean of both)

  F-Measure = (2 * Precision * Recall) / (Precision + Recall)
- 0.0 is poor F-Measure score and 1.0 is best or perfect F-Measure score

Predictive Modelling - Confusion Matrix

- Consider a model that predicts 150 examples for the positive class, 95 are correct (true positives), meaning five were missed (false negatives) and 55 are incorrect (false positives).



Precision = TruePositives / (TruePositives + FalsePositives) = 95 / (95 + 55) = 0.633

Recall = TruePositives / (TruePositives + FalseNegatives) = 95 / (95 + 5) = 0.95

Model has poor precision, but excellent recall.

F-Measure = (2 * Precision * Recall) / (Precision + Recall)

= (2 * 0.633 * 0.95) / (0.633 + 0.95) = (2 * 0.601) / 1.583 = 0.759

- Good recall levels-out poor precision, giving an okay or reasonable F-measure score.

Predictive Modelling - Confusion Matrix

Example: Suppose we are trying to create a model that can predict the result for the disease that is either a person has that disease or not. The table is given for the two-class classifier, which has two predictions "Yes" and "NO." **Yes** defines that patient has the disease, and **No** defines that patient does not has that disease.

The classifier has made a total of 100 predictions. Out of 100 predictions, 89 are true predictions, and 11 are incorrect predictions.

The model has given prediction "yes" for 32 times, and "No" for 68 times. Whereas the actual "Yes" was 27, and actual "No" was 73 times.

In 3 instances, an actual patient was wrongly diagnosed.

Prepare the Confusion matrix and calculate all the performance parameters.

Predictive Modelling - Confusion Matrix

Example: Suppose we are trying to create a model that can predict the result for the disease that is either a person has that disease or not. The table is given for the two-class classifier, which has two predictions "Yes" and "NO." **Yes** defines that patient has the disease, and **No** defines that patient does not has that disease.

The classifier has made a total of 100 predictions. Out of 100 predictions, 89 are true predictions, and 11 are incorrect predictions.

The model has given prediction "yes" for 32 times, and "No" for 68 times. Whereas the actual "Yes" was 27, and actual "No" was 73 times. In 3 instances, an actual patient was wrongly diagnosed.

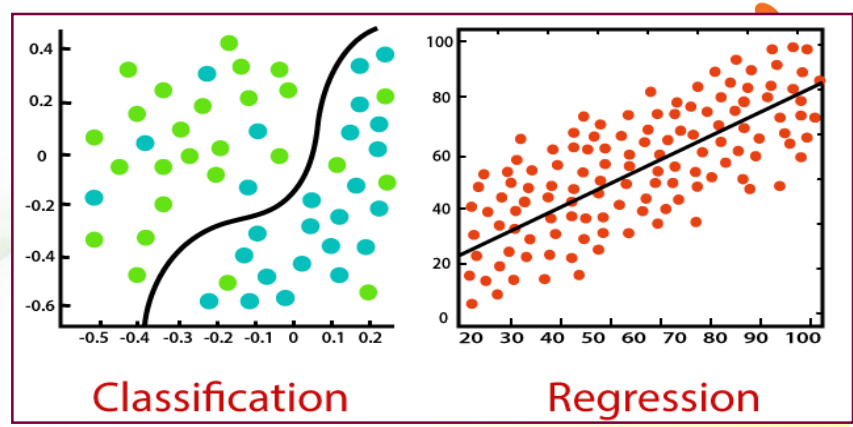
Prepare the Confusion matrix and calculate all the major performance parameters.

n = 100	Actual: No	Actual: Yes	
Predicted: No	TN: 65	FP: 3	68
Predicted: Yes	FN: 8	TP: 24	32
	73	27	

Predictive Modeling

- **Supervised learning**: providing labeled data to machine learning model.
  - labeled dataset is usually data gathered from experience/historical data.
- **Unsupervised learning**: working with unlabeled data.
  - labels in these use cases are often difficult to obtain (not enough knowledge/historical info or labeling is too expensive).
  - lack of labels makes it difficult to set goals for trained model → complicated to measure whether results are accurate.
- **Semi-supervised learning:** Datasets are split into two parts: labeled and unlabeled part.
- **Reinforcement learning:** System learns exclusively from a series of reinforcements.
  - Reinforcements can be positive or negative in relation to a system goal.
  - Positive ones are known as "rewards"; and negative ones as "punishments".

Predictive Modeling



Classification: Process of finding a model/function which helps in separating data into multiple categorical classes.

- Data is categorized under different labels according to some parameters given in input and then labels are predicted for the data.

Regression: Process of finding a model for distinguishing data into continuous real values instead of using classes.

- Helps to identify distribution movement depending on historical data.

- Regression algorithms are used to **predict continuous** values such as price, salary, age, etc.
- Classification algorithms are used to **predict/Classify discrete values** (True or False, Spam or Not Spam, etc.)

Clustering: Process data to find a structure/pattern/cluster in a collection of uncategorized data.

- Hierarchical clustering, K-means clustering

Association: Discovering exciting relationships between variables in large databases.

- Example: Shopping-cart recommendation, video streaming recommendation.

Predictive Modeling

Classification Algorithms:

- Logistic Regression
- K-Nearest Neighbours
- Support Vector Machines
- Naïve Bayes
- Decision Tree Classification
- Random Forest Classification

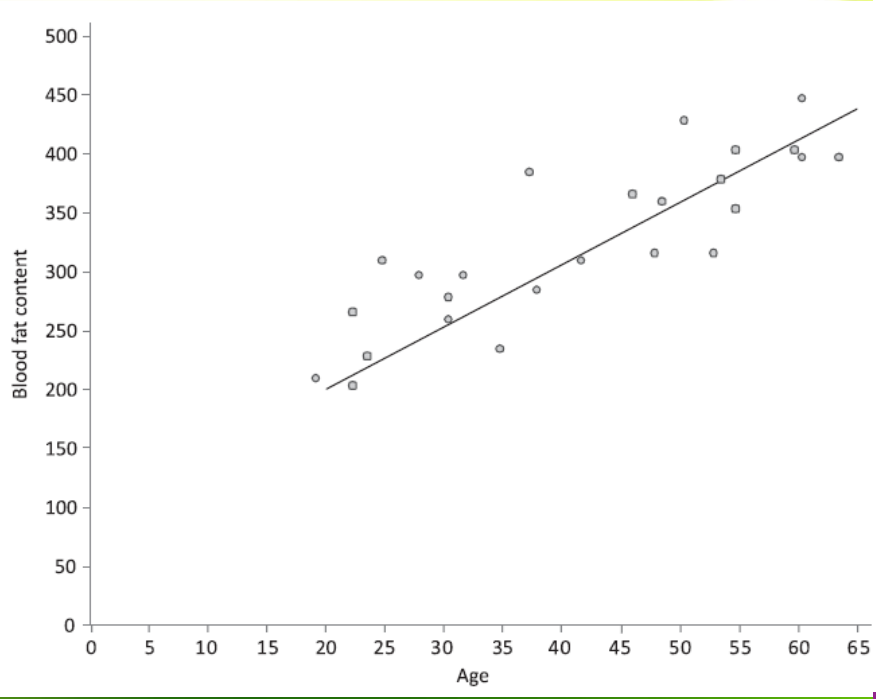
Regression Algorithms:

- Simple Linear Regression
- Multiple Linear Regression
- Polynomial Regression
- Support Vector Regression
- Decision Tree Regression
- Random Forest Regression

Regression Models

Regression: build a function/model that describe relationship between one/more independent variables and a single response variable.

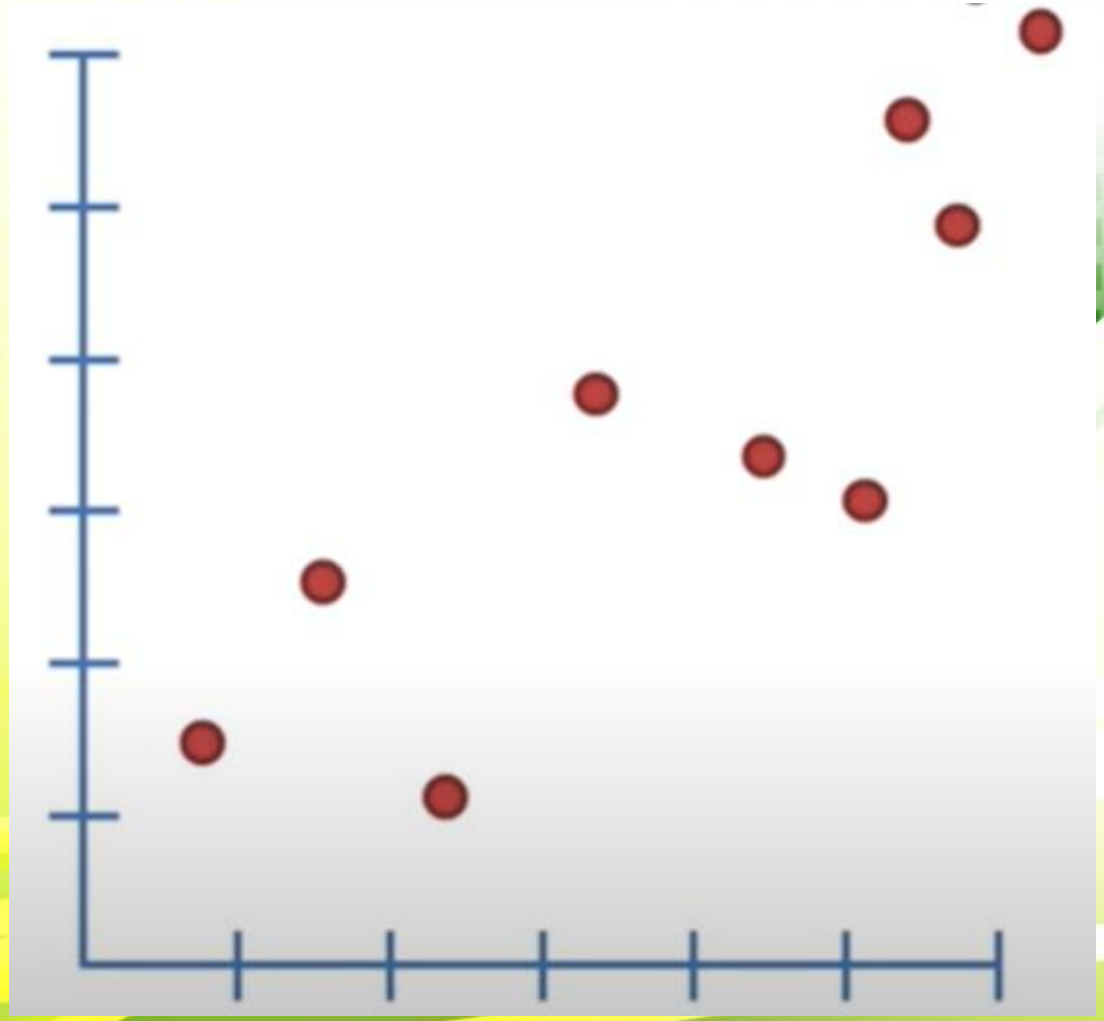
- Plot a graph between the variables which best fit the given data points.
- *Regression shows a line or curve that passes through all data points on a target-predictor graph in such a way that distance between data points and regression line is minimum.*

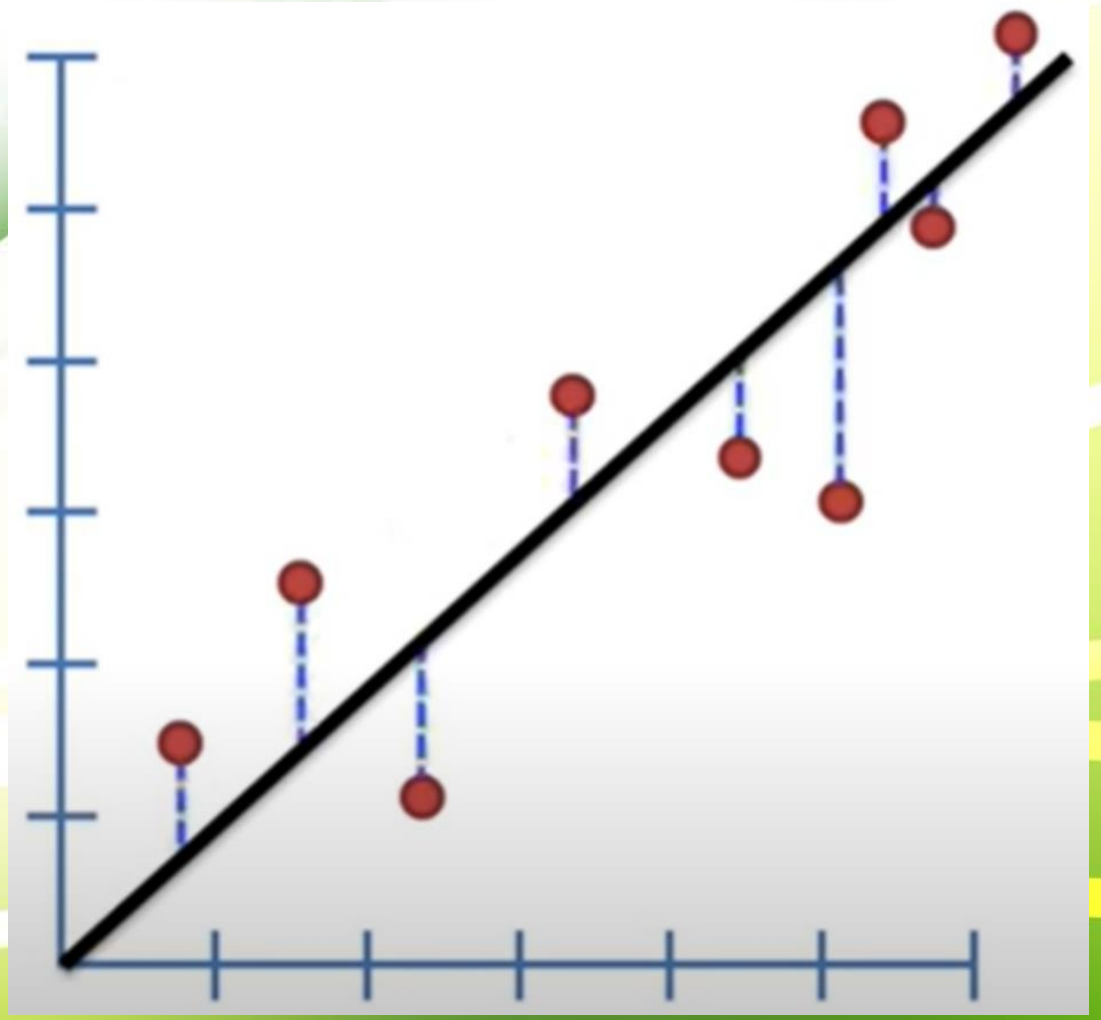


Types of Regression models

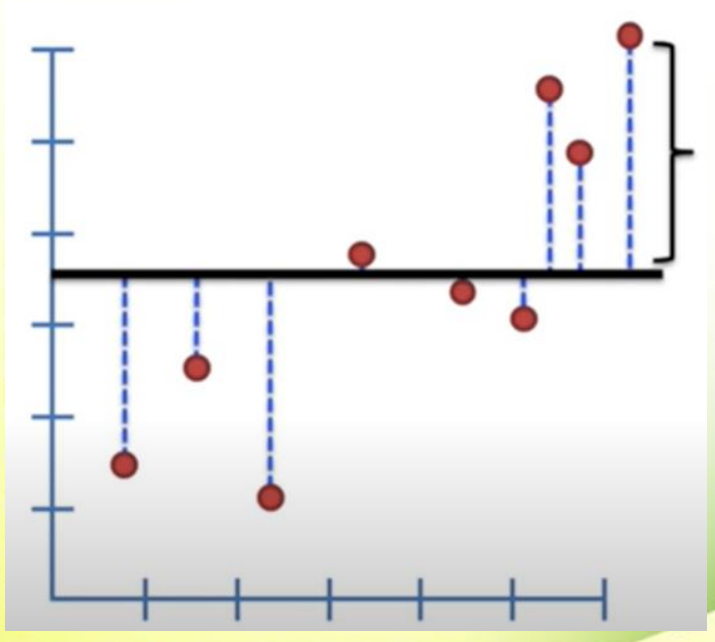
- Linear Regression
- Polynomial Regre6ssion
- Logistic Regression

Linear Regression

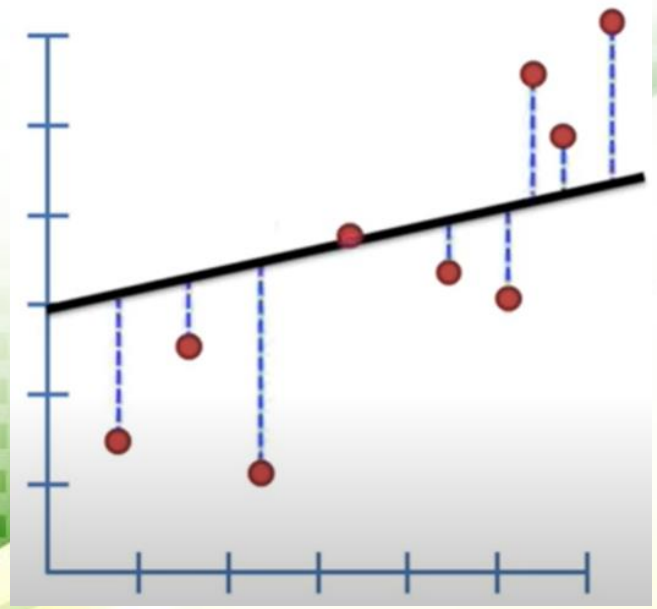




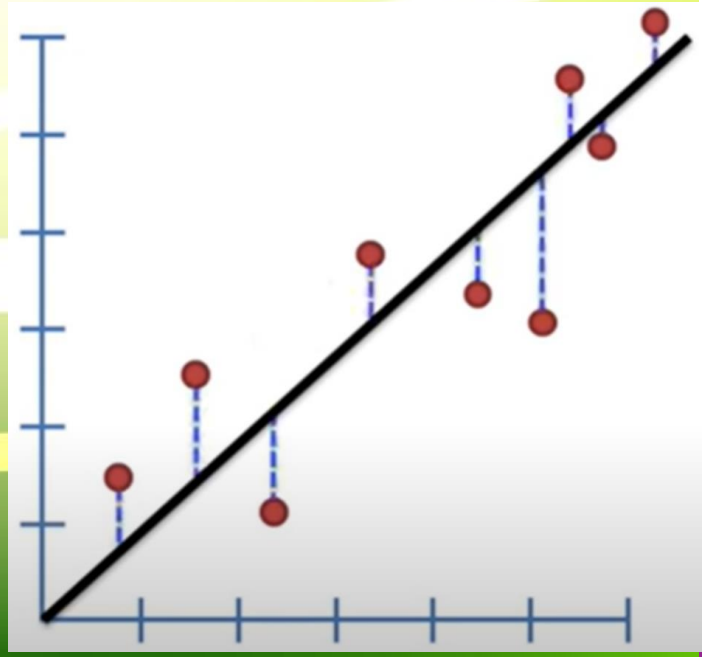
Linear Regression



- Draw a random line through the points.
- Calculate distance between data point & corresponding point on line → Residual.
- Square of least distance.
- Sum of square.



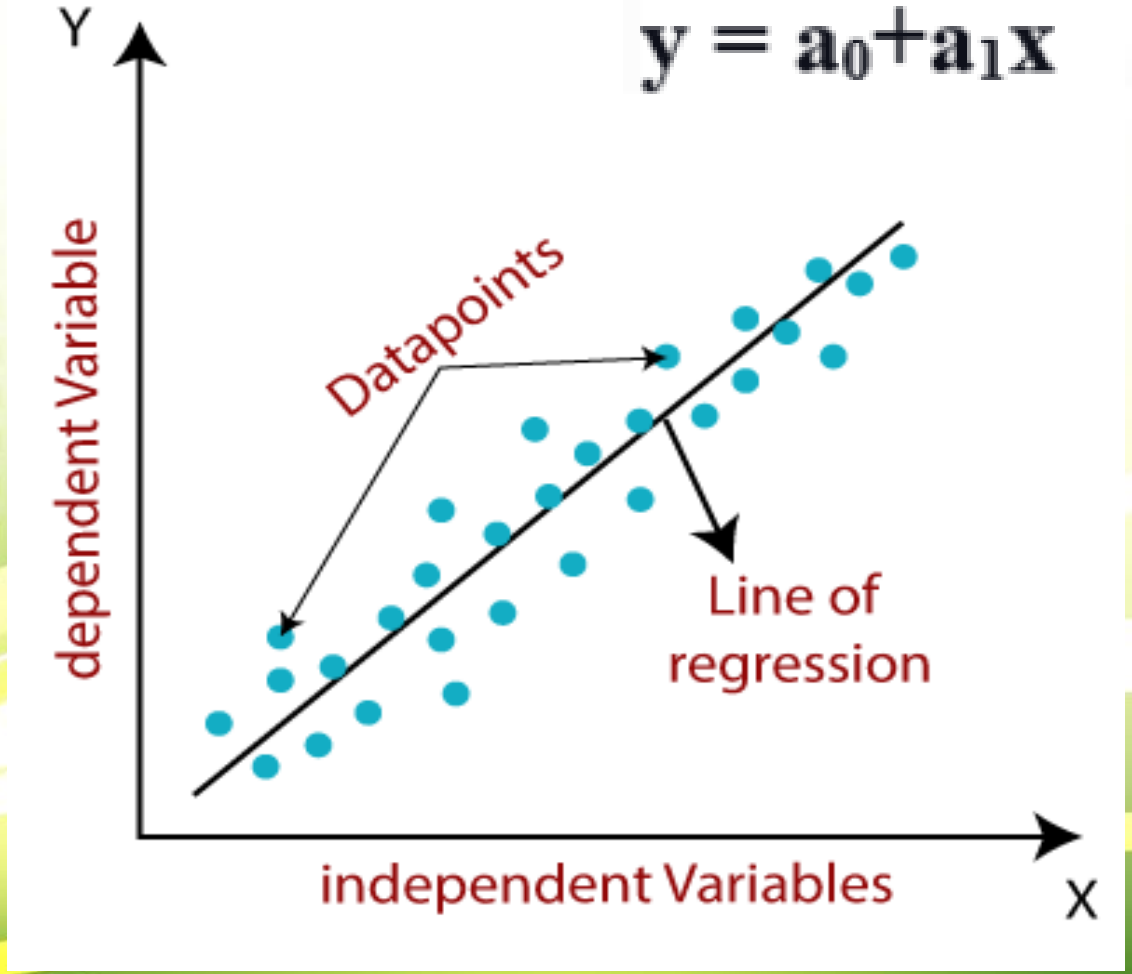
- Rotate the line aiming to reduce error.
- Find each Distance → square → add.



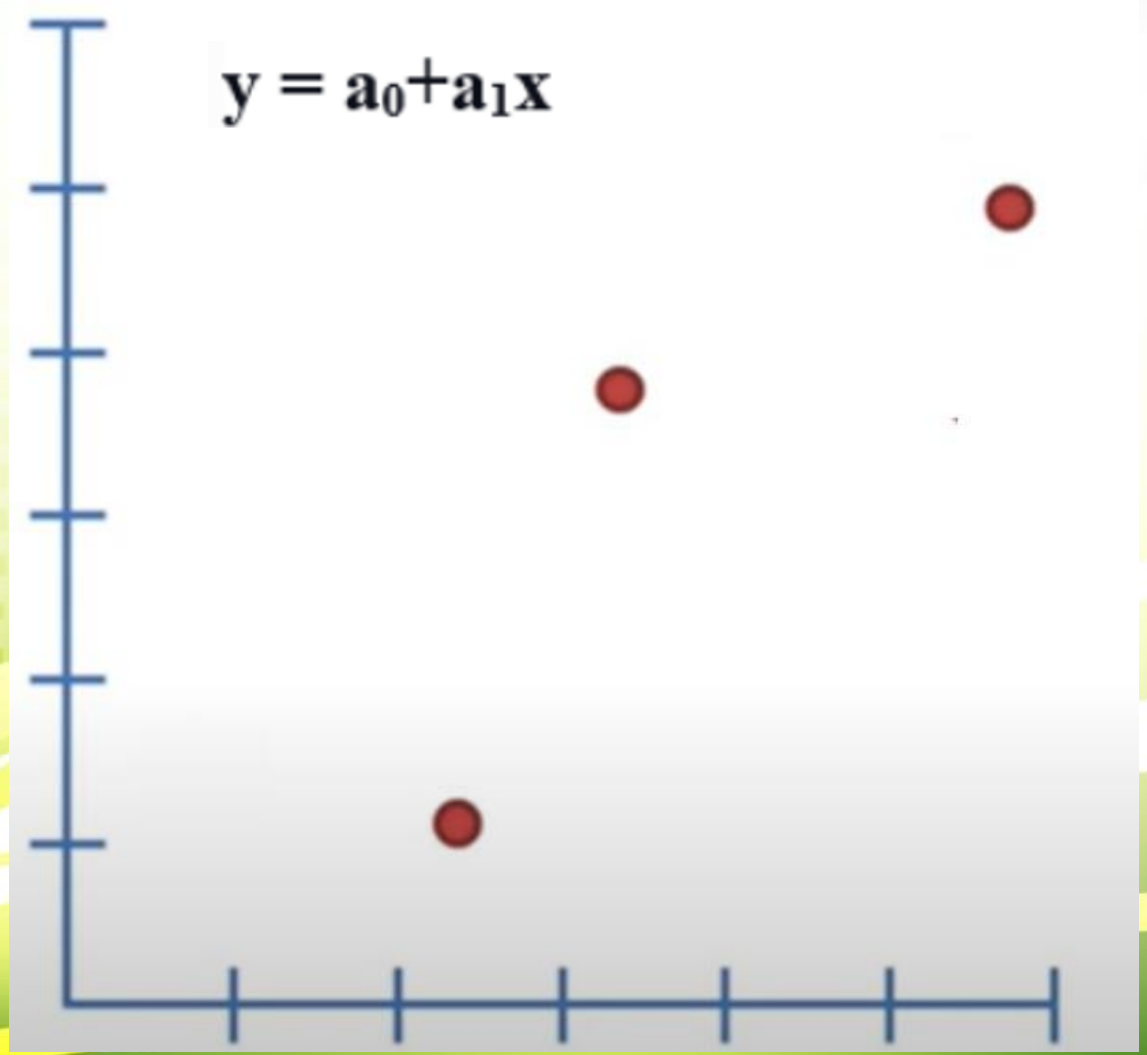
- Regression line
- Lease square distance.

Linear Regression

$$y = a_0 + a_1 x$$

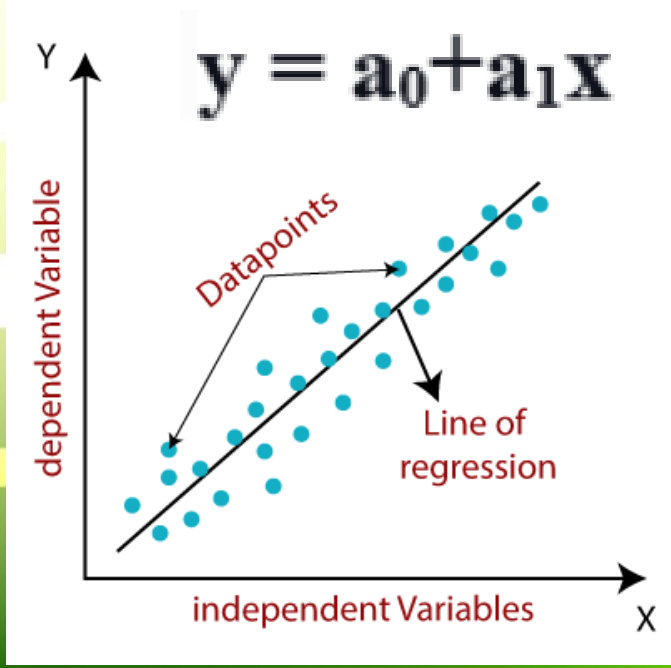


Linear Regression



Linear Regression

- Shows a linear relationship between a dependent (y) and one or more independent (x) variables.
- Finds how value of dependent variable is changing according to value of independent variable.
- Makes predictions for continuous/real or numeric variables; **sales, salary, age, product price,** etc.
- Provides a sloped straight line representing relationship between variables.
- Supervised learning algorithm.
- Goal of linear regression algorithm is to get best values for a_0 and a_1 to find best fit line.
- Best fit line should have least error → error between predicted values and actual values should be minimized.



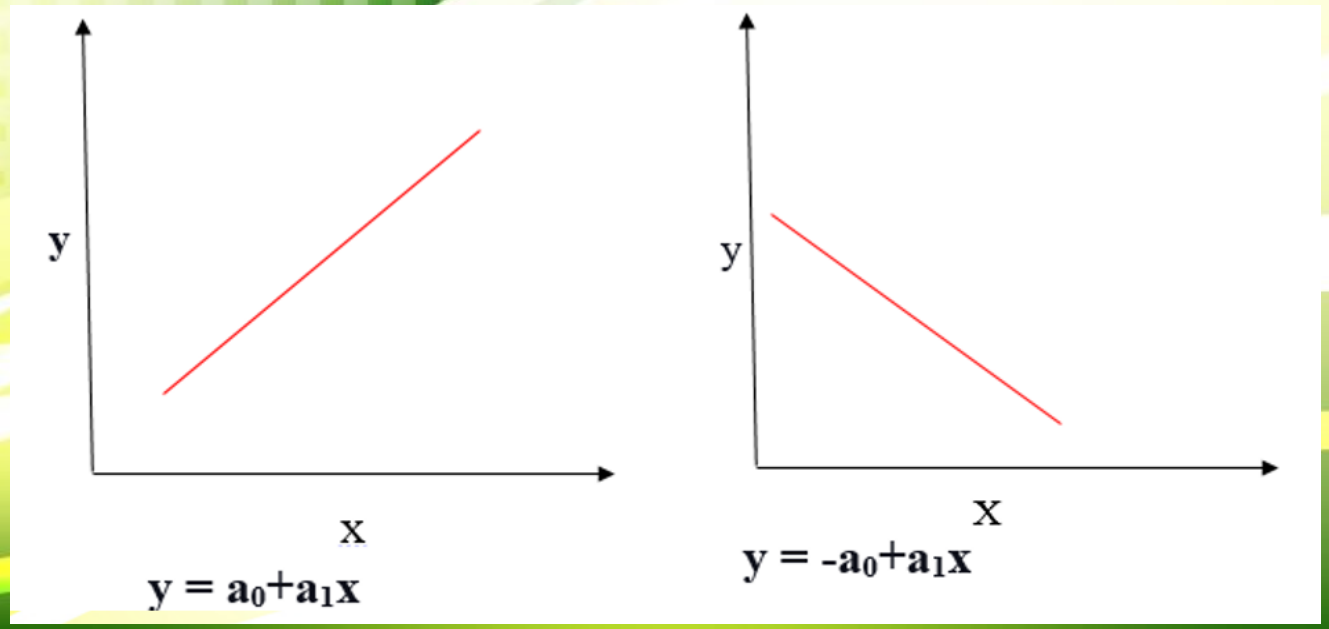
Linear Regression

Positive Linear Relationship

- If dependent variable expands on Y-axis given independent variable progress on X-axis.

Negative Linear Relationship

- If dependent variable decreases on Y-axis given independent variable increases on X-axis.



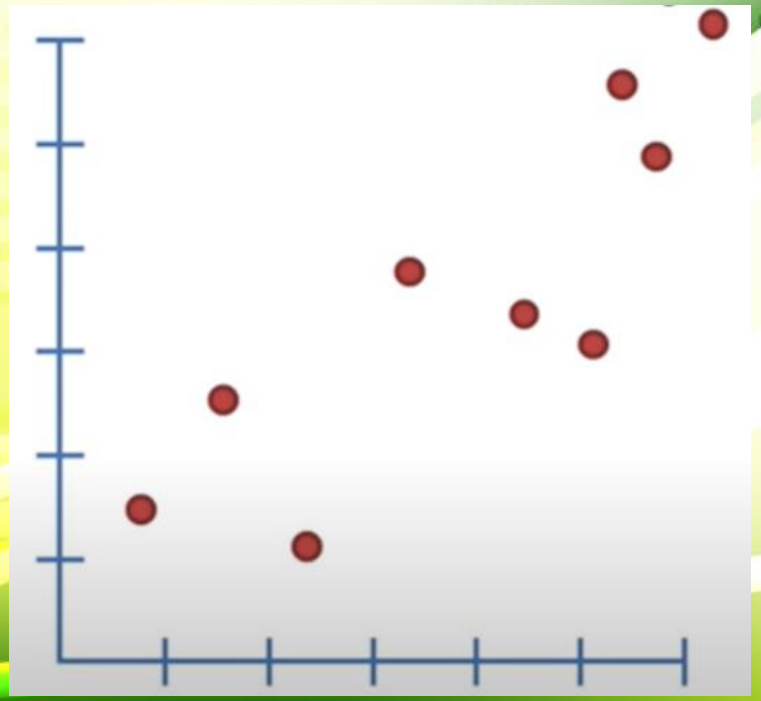
Linear Regression

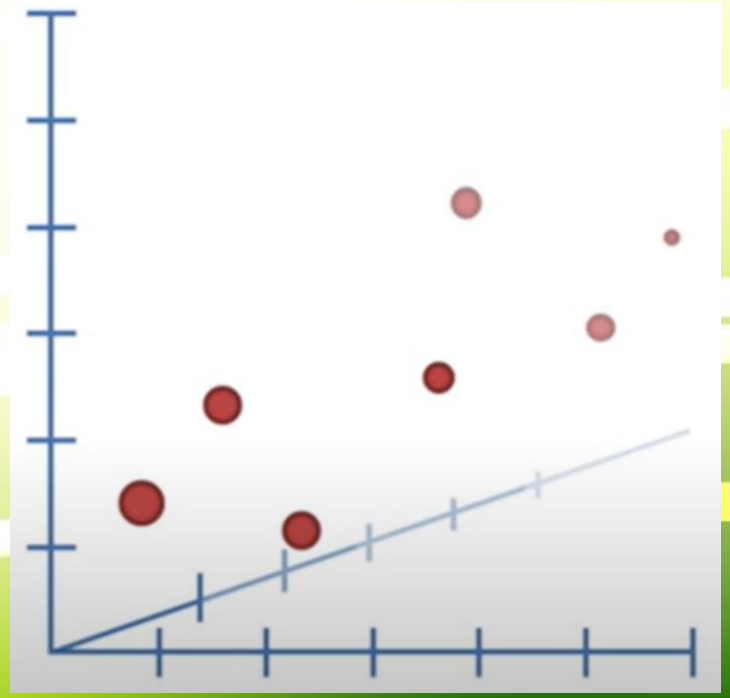
Simple Linear Regression:

- Single independent variable is used to predict value of a numerical dependent variable.

Multiple Linear regression:

- More than one independent variable is used to predict value of a numerical dependent variable.





Linear Regression

- **Simple Linear Regression:**
  - Single independent variable is used to predict value of a numerical dependent variable.
  - $Y = mx + c$
- **Multiple Linear regression:**
  - More than one independent variable is used to predict value of a numerical dependent variable.
  - $Y = a_1x_1 + a_2x_2 + a_3x_3 + a_4x_4 + c$

$$\hat{y} = \hat{b}_0 + \hat{b}_1x_1 + \hat{b}_2x_2 + \cdots + \hat{b}_px_k$$

- a_1, a_2, a_3, a_4 - Regression Coefficient
- capped → estimated coefficient

- $Y = 9 + 3x_1 + 0.9x_2 + 4x_3 + 1.4x_4$
- Regression Coefficient tells which feature(s) have more impact on dependent variable.

Linear Regression

- Size of coefficient for each independent variable tells the impact size of that variable on dependent variable.
- Sign on coefficient (+ or -) tells the direction of effect.

Response

Independent variables

Name	MPG (y)	Cyclinders (x1)	Displacement (x2)	Horsepower (x3)	Weight (x4)	Acceleration (x5)
Chevrolet Chevelle Malibu	18	8	307	130	3504	12
Buick Skylark 320	15	8	350	165	3693	11.5
Plymouth Satellit	18	8	318	150	3436	11
AMC Rebel SST	16	8	304	150	3433	12
Ford Torino	17	8	302	140	3449	10.5

$$\hat{y} = \hat{b}_0 + \hat{b}_1 x_1 + \hat{b}_2 x_2 + \cdots + \hat{b}_p x_k$$

$$y = f(x_i)$$

Model

Linear Regression

Simple Linear Regression:

- b_1 is slope
- b_0 is intercept.

$$y = b_0 + b_1 x$$

$$b_1 = \frac{\sum_{i=1}^{n} (x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^{n} (x_i - \bar{x})^2}$$

$$b_0 = \bar{y} - b_1 \bar{x}$$

Linear Regression

Simple Linear Regression:

mean of x = 39.12
mean of y = 310.72

$$y = b_0 + b_1 x$$

$$b_0 = \bar{y} - b_1 \bar{x}$$

$$b_1 = \frac{\sum_{i=1}^{n} (x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^{n} (x_i - \bar{x})^2}$$

$Slope\,(b_1) = 19{,}157.84/3{,}600.64$

$Slope\,(b_1) = 5.32$

$Intercept\,(b_0) = 310.72 - (5.32 \times 39.12)$

$Intercept\,(b_0) = 102.6$

Blood fat content* = 102.6 + 5.32 × *Age

X	Y	$(x_i - \bar{x})$	$(y_i - \bar{y})$	$(x_i - \bar{x})(y_i - \bar{y})$	$(x_i - \bar{x})^2$
46	354	6.88	43.28	297.7664	47.3344
20	190	−19.12	−120.72	2,308.1664	365.5744
52	405	12.88	94.28	1,214.3264	165.8944
30	263	−9.12	−47.72	435.2064	83.1744
57	451	17.88	140.28	2,508.2064	319.6944
25	302	−14.12	−8.72	123.1264	199.3744
28	288	−11.12	−22.72	252.6464	123.6544
36	385	−3.12	74.28	−231.7536	9.7344
57	402	17.88	91.28	1,632.0864	319.6944
44	365	4.88	54.28	264.8864	23.8144
24	209	−15.12	−101.72	1,538.0064	228.6144
31	290	−8.12	−20.72	168.2464	65.9344
52	346	12.88	35.28	454.4064	165.8944
23	254	−16.12	−56.72	914.3264	259.8544
60	395	20.88	84.28	1,759.7664	435.9744
48	434	8.88	123.28	1,094.7264	78.8544
34	220	−5.12	−90.72	464.4864	26.2144
51	374	11.88	63.28	751.7664	141.1344
50	308	10.88	−2.72	−29.5936	118.3744
34	220	−5.12	−90.72	464.4864	26.2144
46	311	6.88	0.28	1.9264	47.3344
23	181	−16.12	−129.72	2,091.0864	259.8544
37	274	−2.12	−36.72	77.8464	4.4944
40	303	0.88	−7.72	−6.7936	0.7744
30	244	−9.12	−66.72	608.4864	83.1744
			Sum	19,157.84	3,600.64

Linear Regression

Example:

Build the simple linear regression model/function for the data given below.

Age (x)	Sugar Level (Y)
46	354
20	190
52	405
30	263
57	451

$$y = b_0 + b_1 x$$

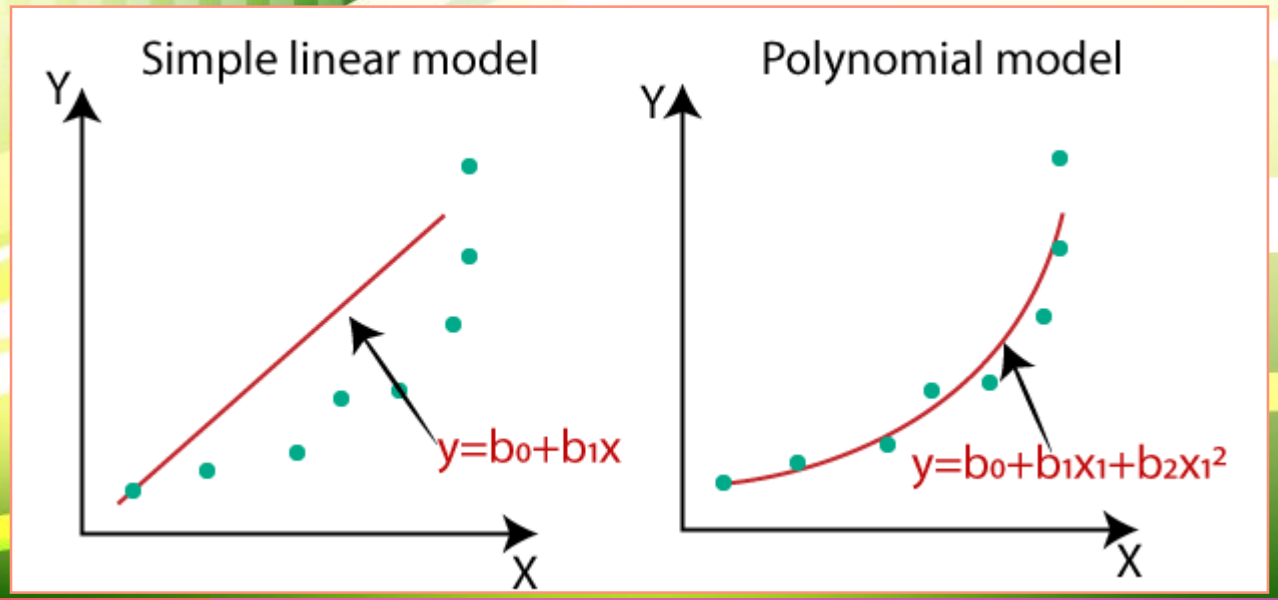
$$b_0 = \bar{y} - b_1 \bar{x}$$

$$b_1 = \frac{\sum_{i=1}^{n} (x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^{n} (x_i - \bar{x})^2}$$

Using this model predict the Sugar level for a new patient of age 39.

Polynomial Regression

- If data points clearly will not fit linear regression (straight line through all data points), it might be ideal for polynomial regression.
- (~~Linear~~)Relationship between variables x and y to find best way to draw a line through data points.
- Special case of Multiple Linear Regression, by adding some polynomial terms to it.
- Relationship between a dependent(y) and independent variable(x) as n^{th} degree polynomial.



Model Assessment

- ***Residual***: error term representing difference between observed value (*y*) and predicted value.

$$\hat{e} = y - \hat{y}$$

- Residual analysis helps to better understand how well model is performing.
- ***Sum of squares total (SST)*** : measure of variation of *y*-values about their mean.
- ***Sum of squares due to regression (SSR):*** differences between predicted/regression values and average *y*-value.
- ***Sum of squares of error (SSE):*** differences between actual *y*-values and predicted *y*-values.

$$SST = \sum_{i=1}^{n} (y_i - \bar{y})^2$$

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

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

$$SST = SSR + SSE$$

Model Assessment

- ***Coefficient of determination* (R^2)**: proportion of variation.

$$SST = \sum_{i=1}^{n} (y_i - \bar{y})^2 \qquad SSR = \sum_{i=1}^{n} (\hat{y}_i - \bar{y})^2 \qquad R^2 = \frac{SSR}{SST}$$

- R^2 values vary between 0 and 1.
- R^2 closer to 1 → more accurate model predictions (models have a *closer fit).*
- In multiple linear regression, *adjusted* R^2 value (R^2 adj) is usually considered to better account for the multiple independent variables used in analysis as well as sample size.

$$R^2_{\text{adj}} = 1 - \left[\frac{(1 - R^2)(n - 1)}{n - k - 1}\right]$$

n : number of observations
k : number of independent variables.

Model Assessment

- *standard error of the estimate* ($S_{y,\,x}$) : measure of variation of *y*-values about regression line.
  - interpreted in a similar manner to standard deviation.
  - indicates model's accuracy: larger the value for standard error of estimate, lower the precision.

$$s_{y.x} = \sqrt{\frac{\text{SSE}}{n-2}}$$

- t-Test, F-Test performed to assess variable dependencies and model performance.
- **Mean Squared Error (MSE):** most common metric for regression models.
- **Mean Absolute Error (MAE):** simple metric; Not preferred where outliers are prominent.
- **Root Mean Squared Error (RMSE):** square root MSE.
  - RMSE penalizes large errors..

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

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|\hat{y}_i - y_i|$$

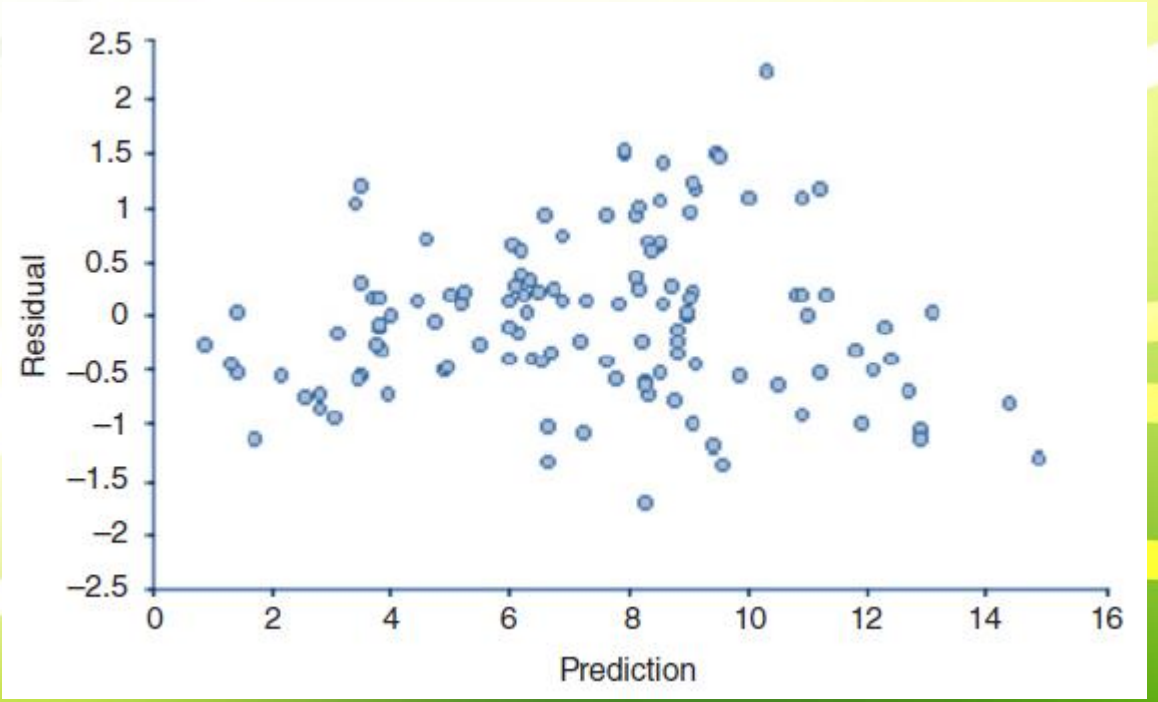
$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(\hat{y}_i - y_i)^2}$$

Model Assessment

- Linear regression models are based on a series of assumptions.
- If a data set does not conform to these assumptions then either the model needs to be adjusted (by applying mathematical **transformation to data**) or particular linear regression not be suitable for modeling that data set.

Assumption - 1:

- ***linearity***: relationship between independent and response variable should be linear.
- A scatterplot of actual response values plotted against predicted values should evenly distribute on both sides of regression line.



Model Assessment

Assumption - 2:

- ***normality of error distribution***: error about the line of regression should be approximately normally distributed for each value of *x*.
  - frequency histogram, statistical measures of skewness/kurtosis, or a normal probability plot.
  - If not a normal distribution, then variable transformations expected.

Assumption - 3:

- ***homoscedasticity of errors***. variation of error/residual across each independent variables should remain constant (as a function of predicted value).

Assumption - 4:

- ***independence of errors***. no trend in residuals based on order in which observations were collected.

Model Assessment

- important to generate simplest possible model that contains only necessary independent variables.
- Ideally number of independent variables should be small and include at least **10 observations** in training set *for every independent variable* included in model. Example:
  - Dataset has 25 independent variables with 300 records.
  - Hypothesis Tests shows 13 important variables.
  - Training set should have minimum 130 records.
- Important to *perform exploratory data analysis to* inspect relationships between variables.
- Perform *transformations* on potential independent variables.
- *Dummy, derived, or composite variables can be generated.*
- Continuous variables may need to be *transformed into a categorical variable.*
- If relationship between a potential independent and response variable needs to be converted from nonlinear to linear, suitable transformations can be used for same.
- Multiple combinations of different independent variables can be used to build set of models from which best performing, most plausible, and simplest model is selected.
- *Standard error*, *t-stat*, and *p-value are* calculated, which can be used to help in selection of independent variables.

Logistic Regression

- Supervised classification algorithm.
- In linear regression problem, target variable(output) y can take only continuous values for a given set of features(inputs) X.
- Logistic regression is popular approach to building models where response variable is categorical.
- Just like Linear regression, it assumes that the data follows a linear function.
- Logistic Regression in its base form is a *Binary Classifier*.
  - Target vector may only take the form of one of two values.
  - Model builds a regression model to predict probability that a given data entry belongs to the category numbered as '1'.
  - A Linear Model, $\beta_0 + \beta_1 x$, is integrated into a Logistic Function (Sigmoid Function).

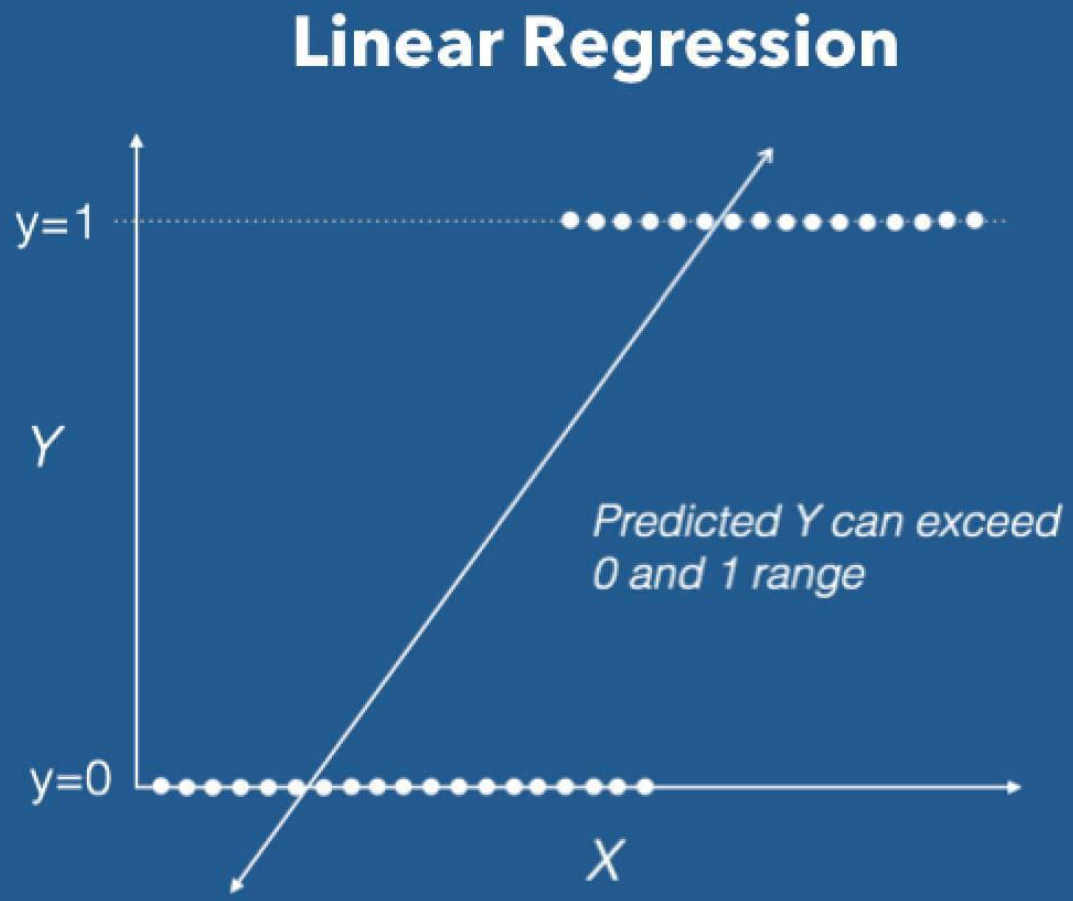
Sugar Level (X)	Diabetes (Y)
354	1
190	0
405	1
263	0
451	1

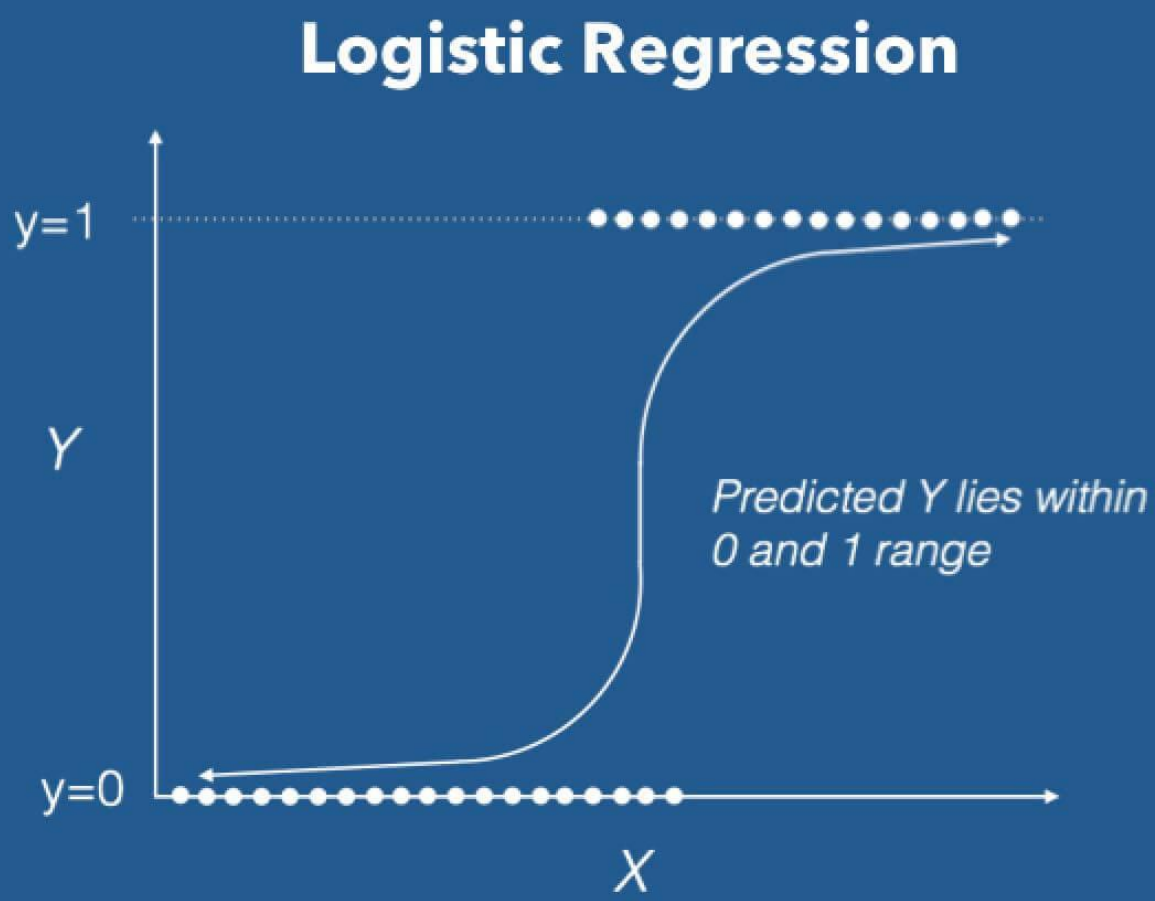
Logistic Regression

Based on the number of categories, Logistic regression can be classified as:

- **binomial:** target variable can have only 2 possible types: “0” or “1”.
  - “win” vs “loss”, “pass” vs “fail”, “dead” vs “alive”, etc.
- **multinomial:** target variable can have 3 or more possible types which are not ordered.
  - “disease A” vs “disease B” vs “disease C”.
- **ordinal:** it deals with target variables with ordered categories.
  - A test score can be categorized: “very poor”, “poor”, “good”, “very good”.

Logistic Regression

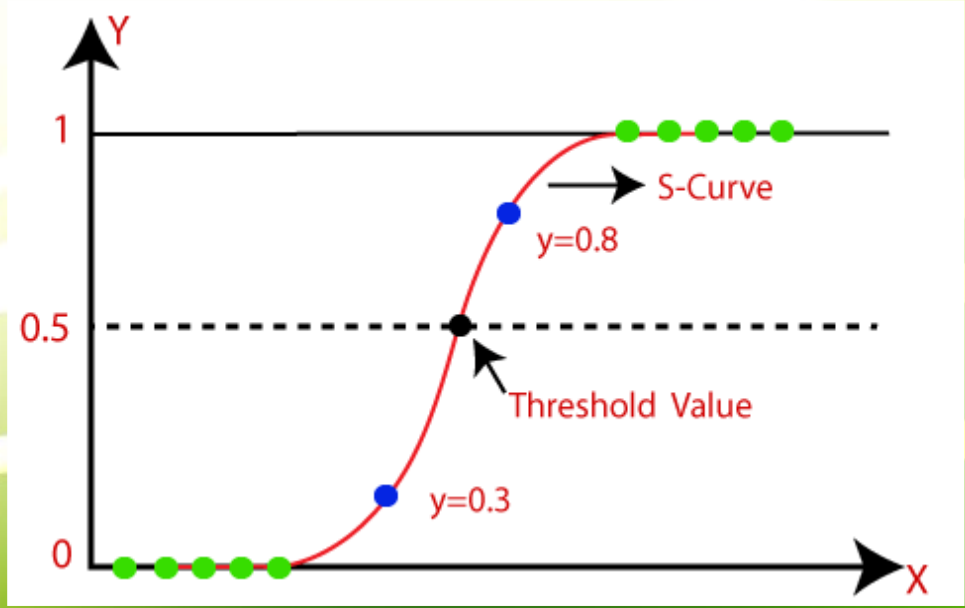




Logistic Regression

Logistic Function (Sigmoid Function):

- A mathematical function used to map the predicted values to probabilities.
- Maps any real value into another value within a range of 0 and 1.
- Value of logistic regression must be between 0 and 1, which cannot go beyond this limit, so it forms a curve like the "S" form.
- S-form curve is called Sigmoid function or logistic function.
- A threshold value defines probability of either 0 or 1.
    - values above threshold value tends to 1,
    - value below the threshold values tends to 0.



Logistic Regression

- standard linear regression formula would compute values outside 0-1 range (not useful)
- Logistic function ensures prediction in 0-1 range

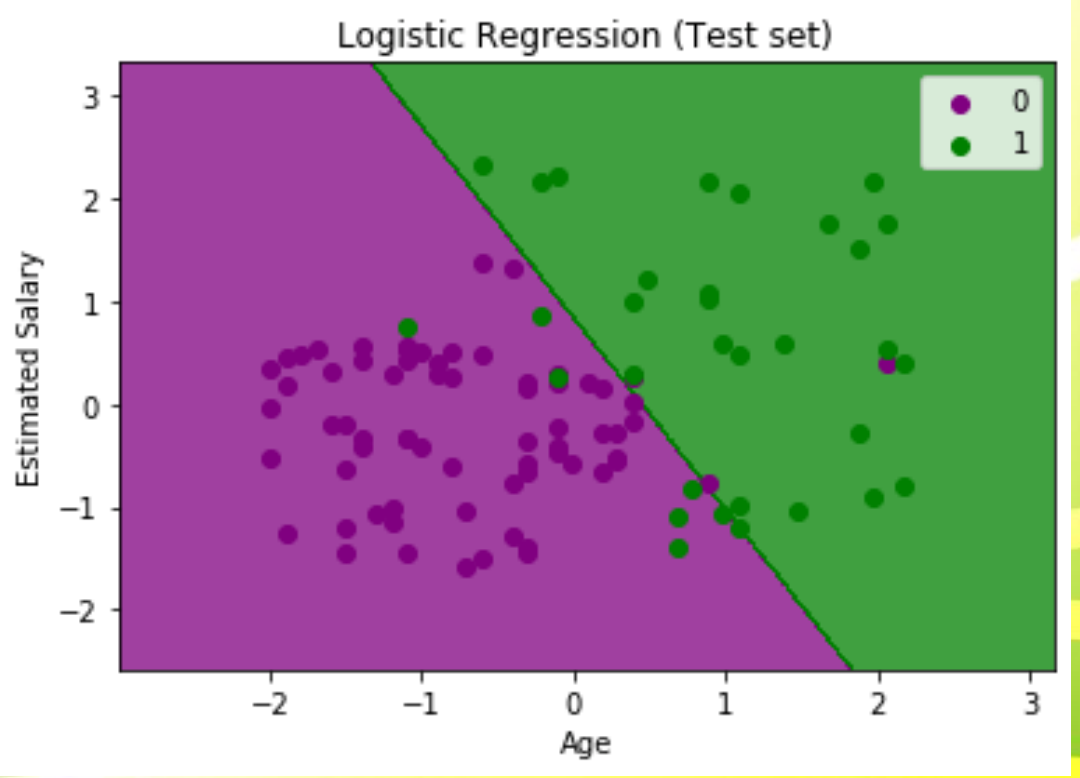
Linear regression/line function $\hat{y} = \hat{b}_0 + \hat{b}_1 x_1 + \hat{b}_2 x_2 + \cdots + \hat{b}_p x_k$

logistic function for response = 1 $P(y = 1) = \frac{1}{1 + e^{(\beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_k x_k)}}$

b_0 is a constant, and b_1 b_k are coefficients to k independent variables (x_1 x_k).

Logistic Regression

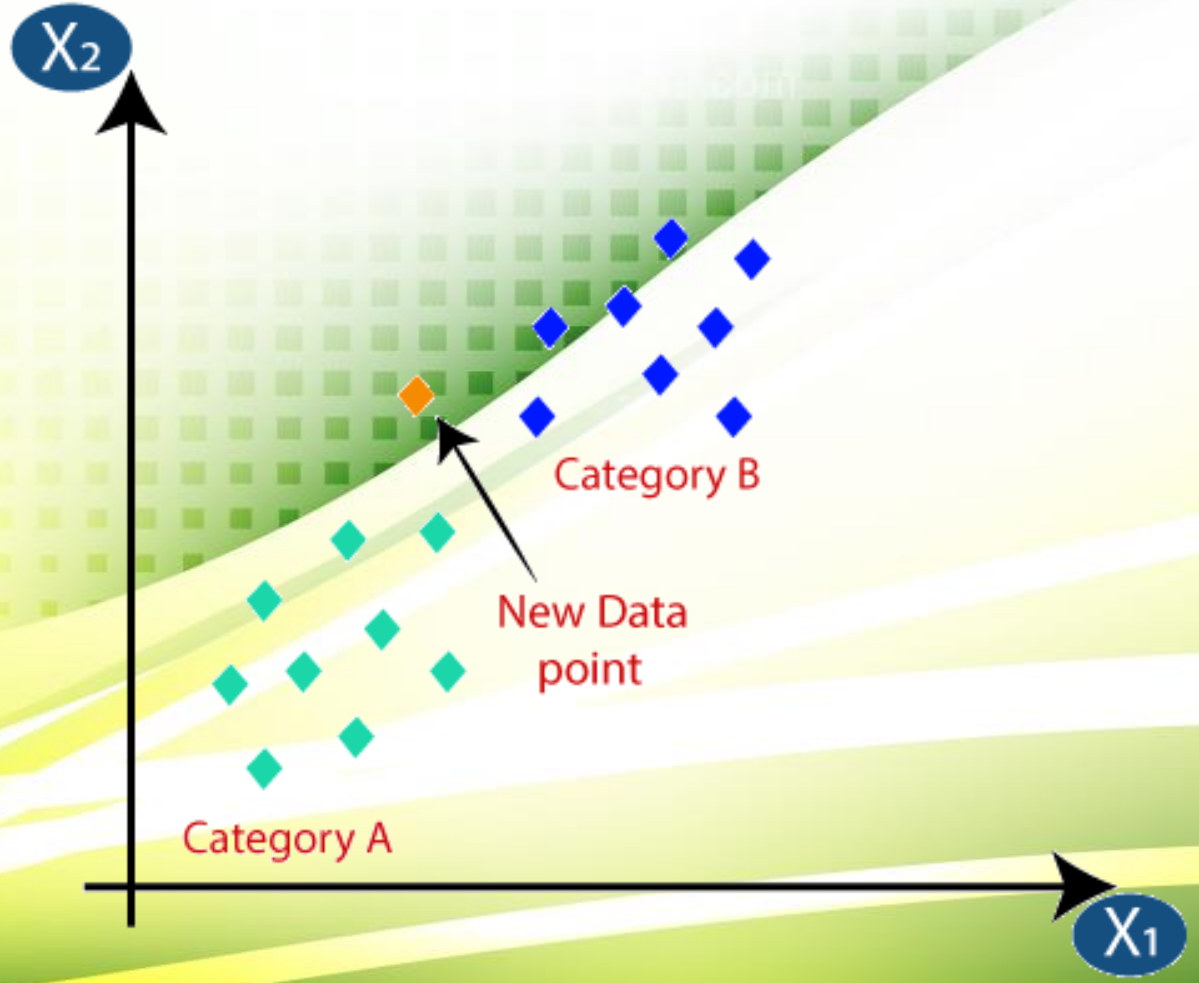
User ID	Gender	Age	EstimatedSalary	Purchased
15624510	Male	19	19000	0
15810944	Male	35	20000	0
15668575	Female	26	43000	0
15603246	Female	27	57000	0
15804002	Male	19	76000	0
15728773	Male	27	58000	0
15598044	Female	27	84000	0
15694829	Female	32	150000	1
15600575	Male	25	33000	0
15727311	Female	35	65000	0
15570769	Female	26	80000	0
15606274	Female	26	52000	0
15746139	Male	20	86000	0
15704987	Male	32	18000	0
15628972	Male	18	82000	0
15697686	Male	29	80000	0
15733883	Male	47	25000	1
15617482	Male	45	26000	1
15704583	Male	46	28000	1
15621083	Female	48	29000	1
15649487	Male	45	22000	1
15736760	Female	47	49000	1



K-Nearest Neighbor (KNN)

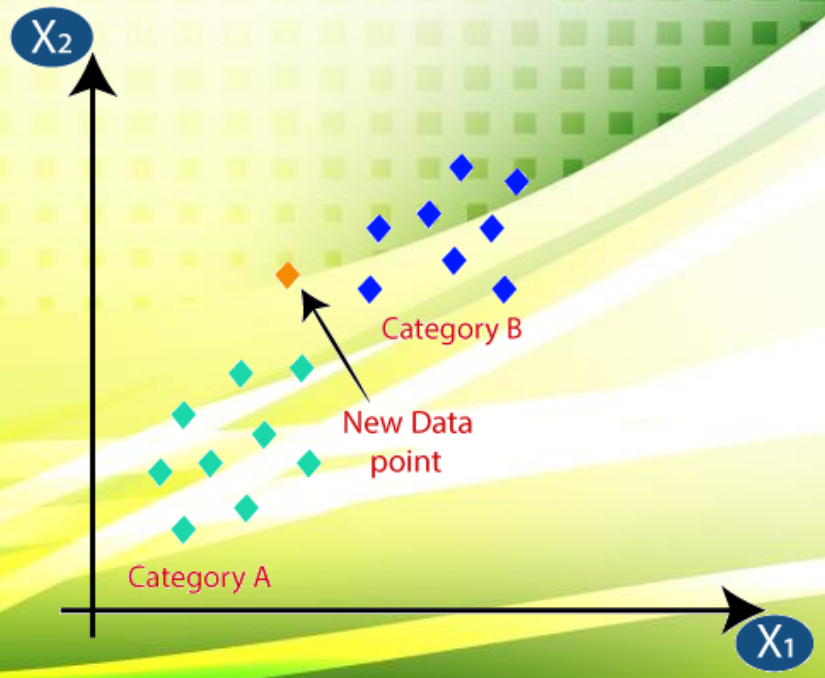
- **Eager learner**: for given training points generalized model construct before performing prediction on new points.
  - Model being ready/prepared, active and eager to classify unobserved data points.
- **Lazy Learning**: no need for learning/training the model and all data points used at time of prediction only.
  - Lazy learner stores merely the training dataset and waits until classification needs to perform.
  - Only when it sees test tuple, starts to perform generalization to classify the tuple based on its similarity to stored training tuples.
- Unlike eager learning methods, lazy learners do less work in training phase and more work in testing phase.
- Lazy learners are also known as **instance-based learners** because lazy learners store training points or instances, and all learning is based on instances.

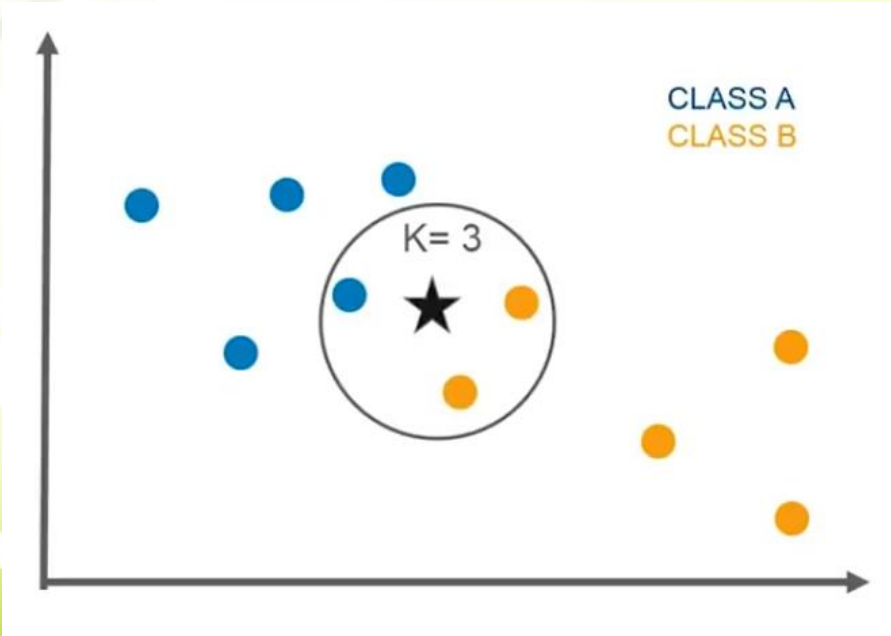
K-Nearest Neighbor (KNN)



K-Nearest Neighbor (KNN)

- Supervised Learning technique.
- Considers K Nearest Neighbors (Data points) to predict class or continuous value for new Datapoint.
  - Considers the similarity between new data and available data, and put new data into the category that is most similar to the available categories.





K-Nearest Neighbor (KNN)

KNN basic steps:

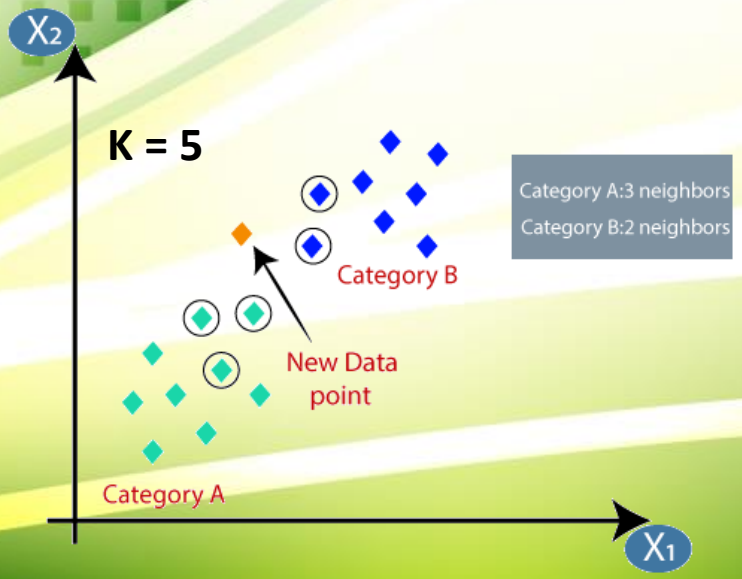
- Calculate distance
- Find closest neighbors
- Vote for labels

K-Nearest Neighbor (KNN)

- **non-parametric algorithm** → does not make any assumption on underlying data.
- Can be used for Regression as well as for Classification.
    - Better suitable for Classification problems.
- **Lazy learner algorithm;** because it does not learn from the training set immediately instead it stores dataset and at the time of classification, it performs an action on the dataset.
    - At training phase **just stores** the dataset and when it gets new data, then it classifies that data into a category that is much similar to new data.
- **Instance-based learning**: Do not learn weights from training data to predict output, but use entire training instances to predict output for unseen data.

K-Nearest Neighbor (KNN)

- **Step-1:** Select the number K of the neighbors
- **Step-2:** Calculate the *Euclidean distance* of new data from all datapoints.
- **Step-3:** Take the K nearest neighbors as per calculated Euclidean distance (nearest / shorted distance).
- **Step-4:** Among these k neighbors, vote/count the number of data points in each category.
- **Step-5:** Assign new data points to that category for which number of neighbor is maximum.
- **Step-6:** model is ready.



K-Nearest Neighbor (KNN)

- Three distance measures valid for continuous variables.
- Hamming distance for categorical variables.
- Standardization of numerical variables between 0 and 1 when there is mixture of numerical and categorical variables in dataset.

Distance functions

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

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

Minkowski $\left(\sum_{i=1}^{k}(|x_i - y_i|)^q\right)^{1/q}$

$d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$

$|x_1 - x_2| + |y_1 - y_2|$

Hamming Distance

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

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

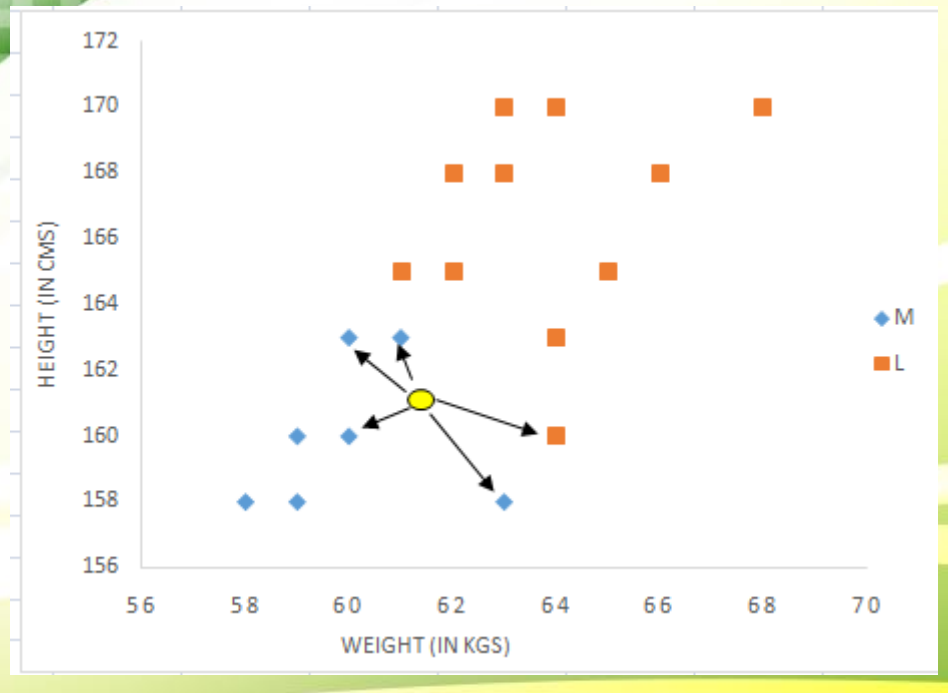
$$x \neq y \Rightarrow D = 1$$

X	Y	Distance
Male	Male	0
Male	Female	1

K-Nearest Neighbor (KNN)

Example

Height (in cms)	Weight (in kgs)	T Shirt Size	Distance	
158	58	M	4.2	
158	59	M	3.6	
158	63	M	3.6	
160	59	M	2.2	3
160	60	M	1.4	1
163	60	M	2.2	3
163	61	M	2.0	2
160	64	L	3.2	5
163	64	L	3.6	
165	61	L	4.0	
165	62	L	4.1	
165	65	L	5.7	
168	62	L	7.1	
168	63	L	7.3	
168	66	L	8.6	
170	63	L	9.2	
170	64	L	9.5	
170	68	L	11.4	
161	61			



K-Nearest Neighbor (KNN)

Standardization

- height in cms (large) will influence more on distance calculation compared to weight in kgs.
- important to standardize variables before calculating distance

$$Xs = \frac{X - mean}{s.d.}$$

$$Xs = \frac{X - mean}{max - min}$$

$$Xs = \frac{X - min}{\max - min}$$

Height (in cms)	Weight (in kgs)	T Shirt Size	Distance	
158	58	M	4.2	
158	59	M	3.6	
158	63	M	3.6	
160	59	M	2.2	3
160	60	M	1.4	1
163	60	M	2.2	3
163	61	M	2.0	2
160	64	L	3.2	5
163	64	L	3.6	
165	61	L	4.0	
165	62	L	4.1	
165	65	L	5.7	
168	62	L	7.1	
168	63	L	7.3	
168	66	L	8.6	
170	63	L	9.2	
170	64	L	9.5	
170	68	L	11.4	
161	61			

	A	B	C	D	E
1	Height (in cms)	Weight (in kgs)	T Shirt Size	Distance	
2	-1.39	-1.64	M	1.3	
3	-1.39	-1.27	M	1.0	
4	-1.39	0.25	M	1.0	
5	-0.92	-1.27	M	0.8	4
6	-0.92	-0.89	M	0.4	1
7	-0.23	-0.89	M	0.6	3
8	-0.23	-0.51	M	0.5	2
9	-0.92	0.63	L	1.2	
10	-0.23	0.63	L	1.2	
11	0.23	-0.51	L	0.9	5
12	0.23	-0.13	L	1.0	
13	0.23	1.01	L	1.8	
14	0.92	-0.13	L	1.7	
15	0.92	0.25	L	1.8	
16	0.92	1.39	L	2.5	
17	1.39	0.25	L	2.2	
18	1.39	0.63	L	2.4	
19	1.39	2.15	L	3.4	
20					
21	-0.7	-0.5			

K-Nearest Neighbor (KNN)

Select value of K

- No optimal number of neighbors suits all kind of data sets.
- Each dataset has it's own requirements.
- At low K values, there is **overfitting** of data (high variance → test error is high and train error is low).
  - For small K, noise will have higher influence on result.
  - As we increase the value for K, the test error is reduced.
- Large values for K are good, but it may find some difficulties.
  - After a certain K value, bias/ **underfitting** is introduced and test error goes high.
  - Large K make it computationally expensive.
- No particular way to determine the best value for "K".
  - Need to try **some** values to find the best out of them (**Elbow method**)
  - Most preferred value of K to start trying is 5.
- K value should be odd while considering binary (two-class) classification.

K-Nearest Neighbor (KNN)

KNN regression is same like KNN classification.

Age	Loan	House Price Index	Distance	
25	$40,000	135	102000	
35	$60,000	256	82000	
45	$80,000	231	62000	
20	$20,000	267	122000	
35	$120,000	139	22000	2
52	$18,000	150	124000	
23	$95,000	127	47000	
40	$62,000	216	80000	
60	$100,000	139	42000	3
48	$220,000	250	78000	
33	$150,000	264	8000	1
48	**$142,000**	**?**		

$$D = \sqrt{(x_1 - y_1)^2 + (x_2 - y_2)^2}$$

K=3

Prediction for HPI by averaging K neighbors values

HPI = (264+139+139)/3 = 180.7

K-Nearest Neighbor (KNN)

KNN regression (*with Standardization*)

Age	Loan	House Price Index	Distance
0.125	0.11	135	0.7652
0.375	0.21	256	0.5200
0.625	0.31	231	0.3160
0	0.01	267	0.9245
0.375	0.50	139	0.3428
0.8	0.00	150	0.6220
0.075	0.38	127	0.6669
0.5	0.22	216	0.4437
1	0.41	139	0.3650
0.7	1.00	250	0.3861
0.325	0.65	264	0.3771
0.7	**0.61**	**?**	

$$X_s = \frac{X - Min}{Max - Min}$$

~~HPI = (264+139+139)/3 = 180.7~~

K=3
Prediction for HPI by averaging K neighbors values

HPI = (231+139+139)/3 = **169.7**

K-Nearest Neighbor (KNN)

Advantages of KNN Algorithm:

- simple to understand and implement.
- robust to the noisy training data
- No assumptions about data
- Can be applied to both classification and regression
- Works easily on multi-class problems

Disadvantages of KNN Algorithm:

- Always needs to determine the best value of K, which may be complex some time.
- computation cost is high because of calculating distance between the data points for all training samples.
- Sensitive to scale of data
- Struggle when high number of independent variables

K-Nearest Neighbor (KNN)

Example

Weight (in Kg) X	Height (in cm) Y
58	158
59	158
63	158
59	160
60	160
60	163
61	163
64	160
64	163
61	165

For X = 63, which model will you apply;

KNN Regressor or Simple linear Regressor?

(use K = 5)

K-Nearest Neighbor (KNN)

Example

Weight (in Kg)	Height (in cm)	T shirt Size	T shirt Size
58	158	28	S
59	158	28.5	S
63	158	28.5	S
59	160	29	M
60	160	29	M
60	163	29	M
61	163	29.5	M
64	160	30	L
64	163	30	L
61	165	29.5	M

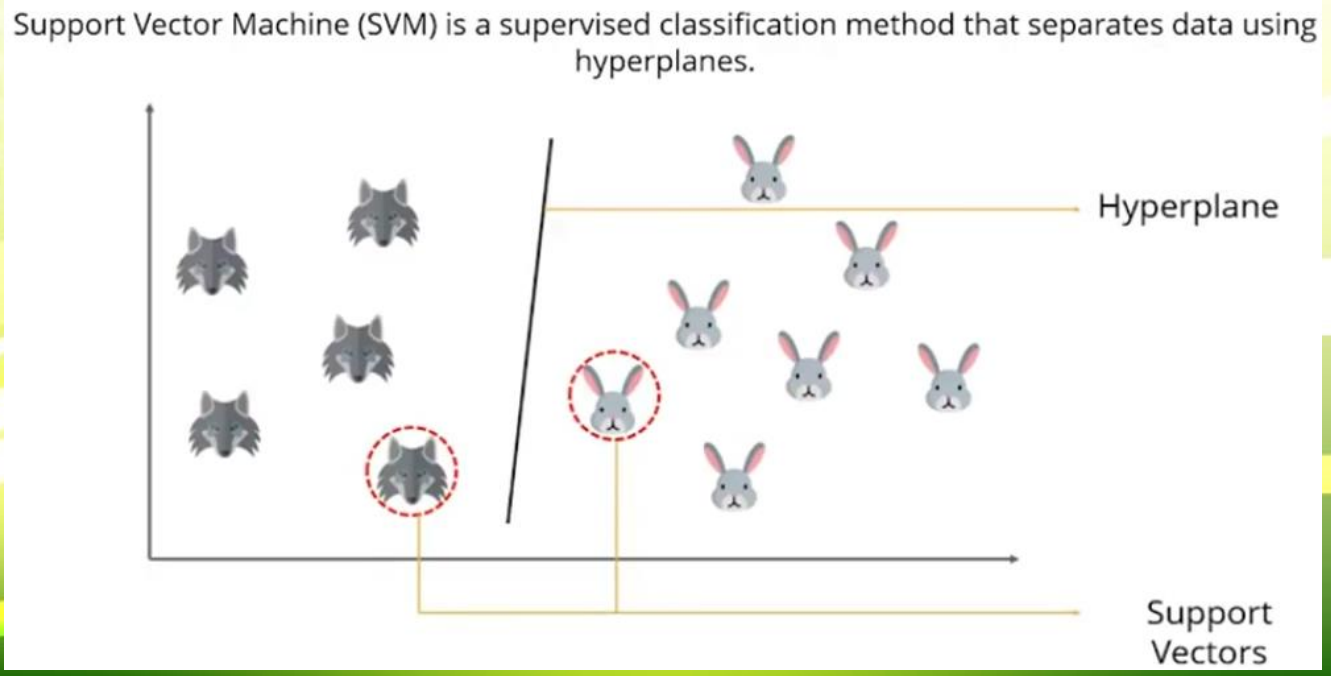
Does K=4 & K=9 give same result in prediction of the T-shirt Size for a persion with Height 159 and Weight 60.

Apply (i) KNN Classifier,

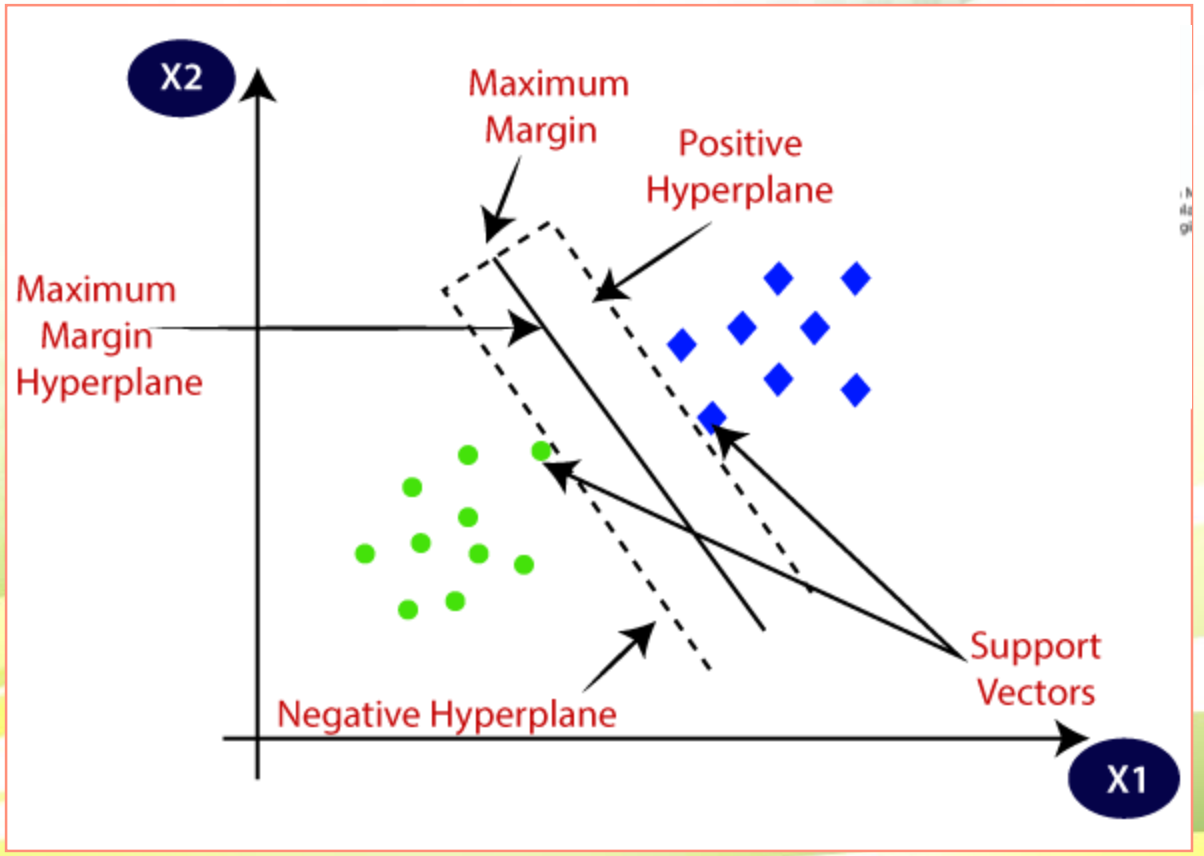
(ii) KNN Regressor

Support Vector Machine (SVM)

- Supervised learning models; Used for **classification** and regression analysis
- Discriminative classifier formally defined by a separating hyperplane.
  - Given a labeled training data, the algorithm outputs an optimal hyperplane.
- In addition to performing linear classification, SVMs can efficiently perform a non-linear classification, implicitly mapping their inputs into high-dimensional feature spaces.



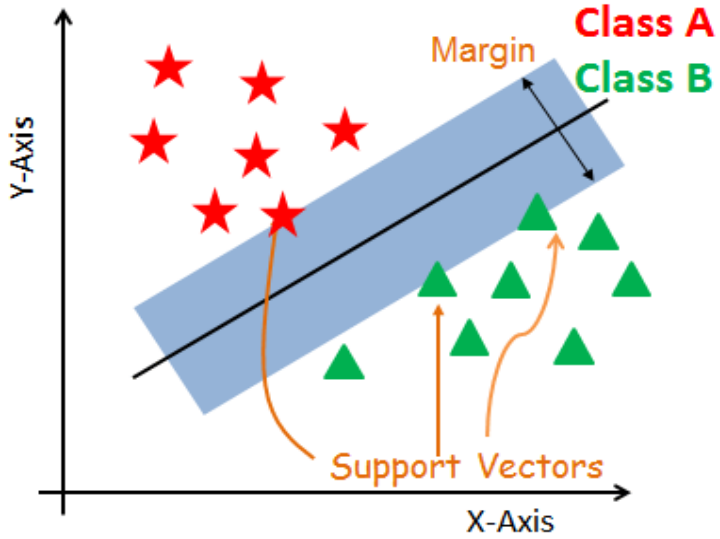
Support Vector Machine (SVM)



Support Vector Machine (SVM)

Support Vectors

- Support vectors are the data points, which are closest to the hyperplane.
- These points define the separating line better by calculating margins.
- These points are more relevant to the construction of the classifier.



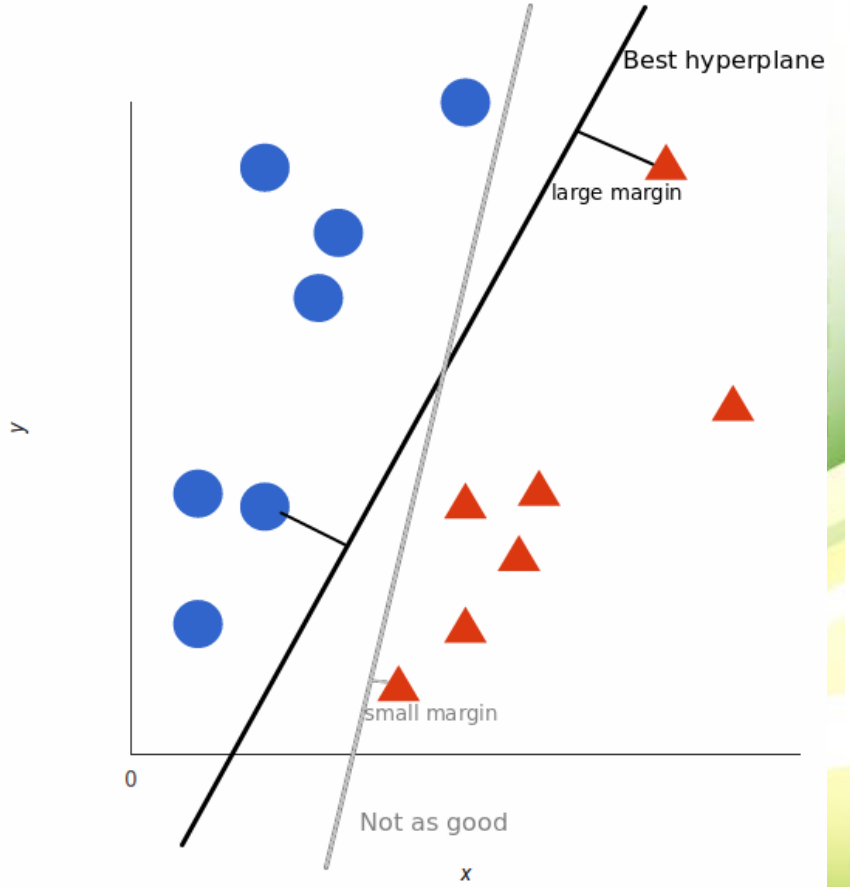
Hyperplane

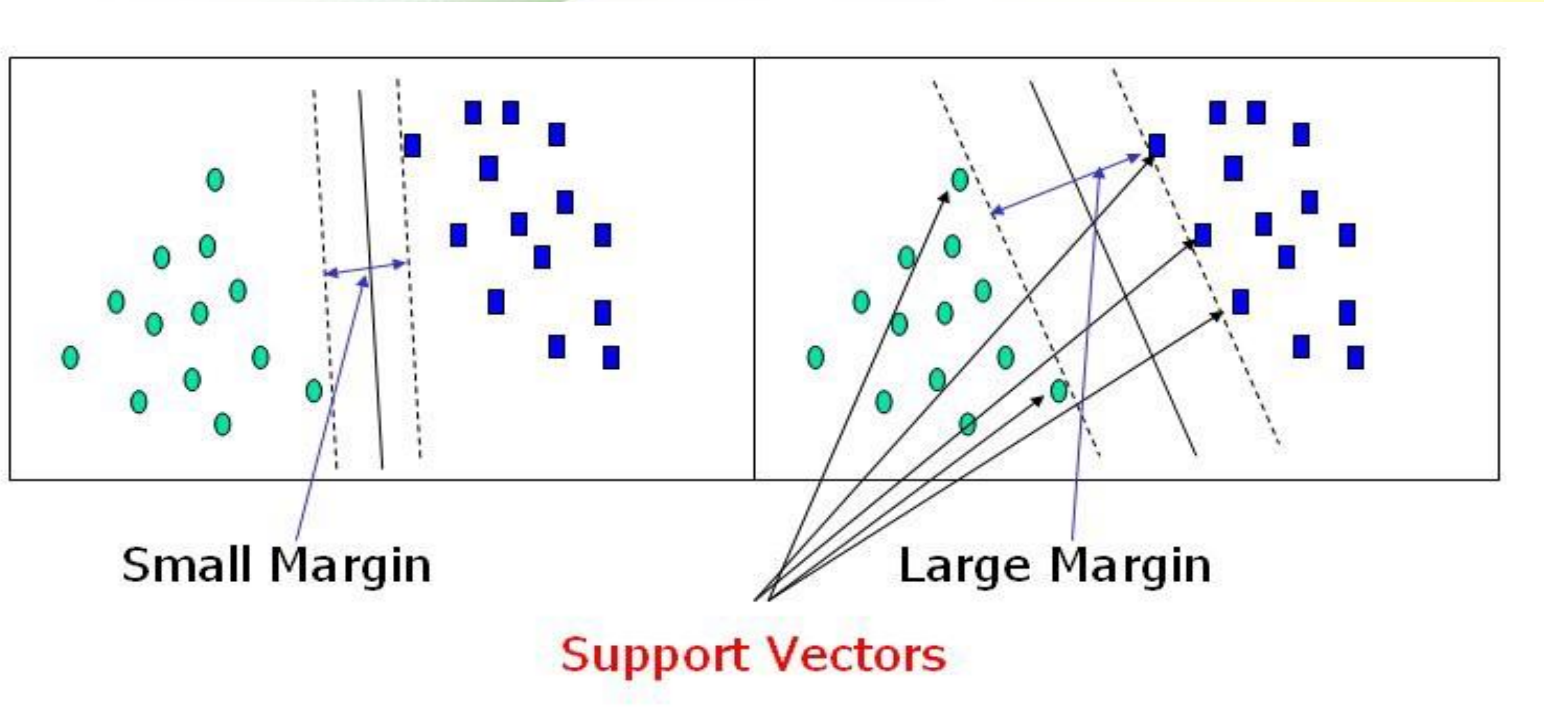
- Hyperplane acts as a decision plane which separates between a set of objects having different class memberships.

Margin

- Gap between the two lines on the closest class points.
- Calculated as the perpendicular distance from the line to support vectors or closest points.
- If the margin is larger in between the classes, then it is considered a good margin, a smaller margin is a bad margin.

Support Vector Machine (SVM)

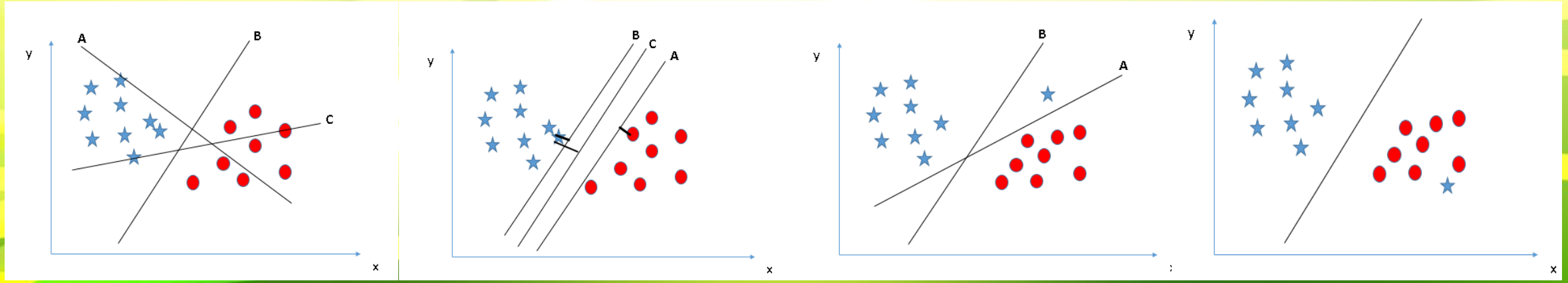




Support Vector Machine (SVM)

Identify the right hyper-plane:

- Select the hyper-plane which segregates the two classes better..
- If multiple hyper-planes are segregating the properly classes; then maximizing the distances (**Margin)** between nearest data point (either class) and hyper-plane will help to decide best hyper-plane.
- Classification accuracy carries more importance over Margin as a measure.
- SVM algorithm has a feature to ignore outliers and find the hyper-plane that has the maximum margin (if not all).
- If linear hyper-plane between the two classes is impossible, a non-linear way can be considered.



Support Vector Machine (SVM)

- In ML, cost functions are **used to estimate how badly models are performing**.
- Cost function is a measure of how wrong model is in terms of its ability to estimate the relationship between X and Y.
- Typically expressed as difference/distance between predicted value and actual value.

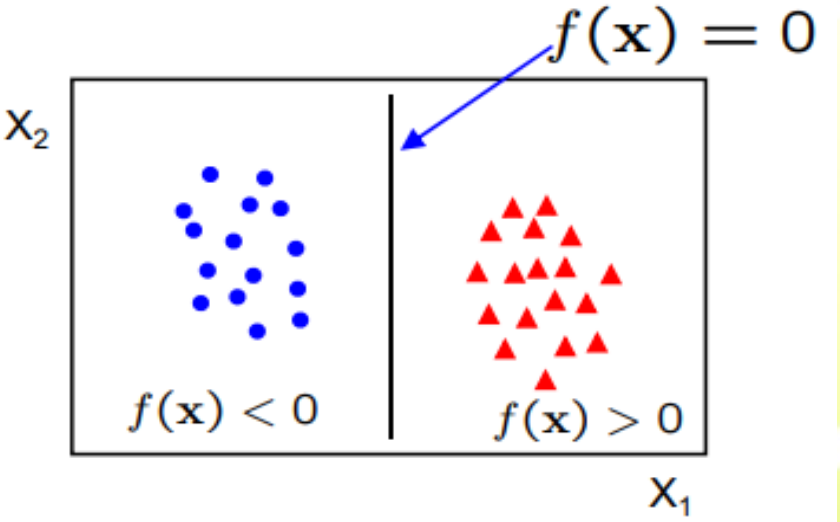
- In SVM algorithm, aim is to maximize margin between data points and hyperplane.
- Loss function helps maximize this margin.

Support Vector Machine (SVM)

- Function that defines hyperplane in linear SVM

$$f(X) = w^T * X + b$$

  - **w :** weight vector that needs to be minimized,
  - **X :** data to classify,
  - **b :** linear coefficient estimated from training data.



- Polynomial hyperplane function

$$f(X1, X2) = (a + X1^T * X2) ^ b$$

Support Vector Machine (SVM)

- used for *Face detection, image classification, text categorization*, etc.

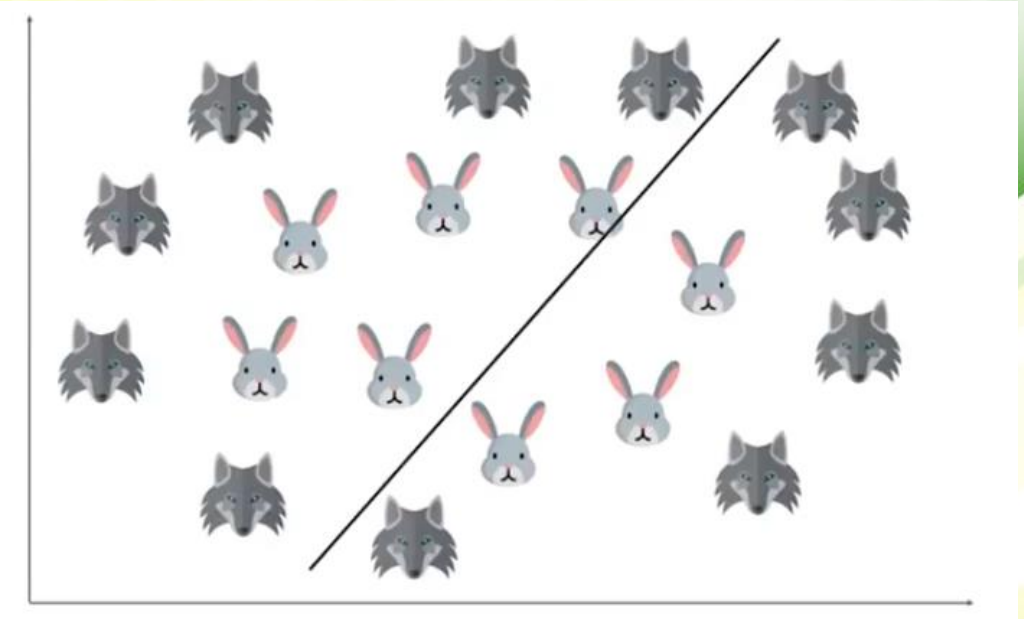
SVM can be of two types:

- **Linear SVM:** used for linearly separable data.
  - dataset can be classified into two classes by using a single straight line → linearly separable data.

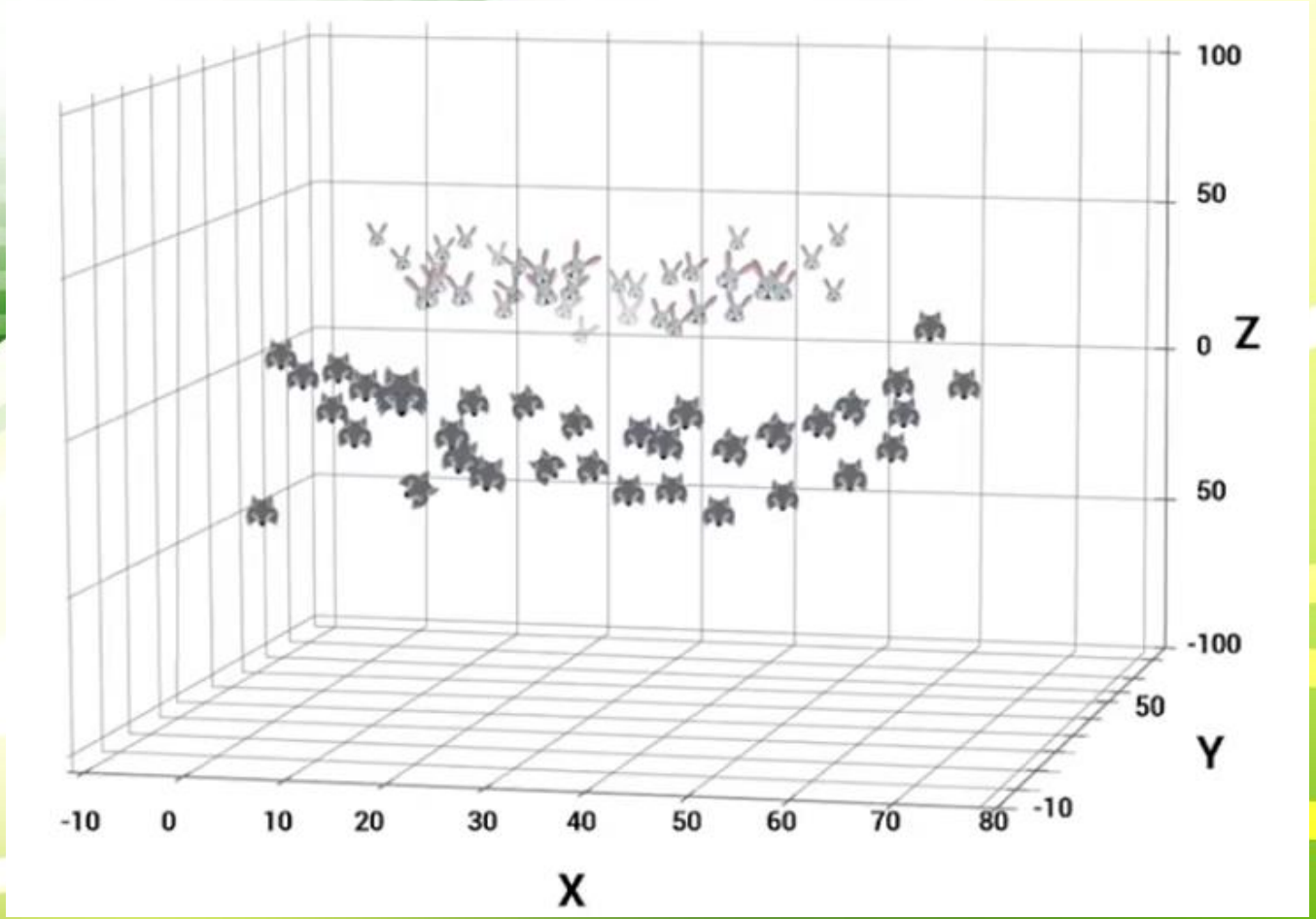
- **Non-linear SVM:** used for non-linearly separated data.
  - dataset cannot be classified by using a straight line.

Support Vector Machine (SVM)

Linear SVM



Non-linear SVM

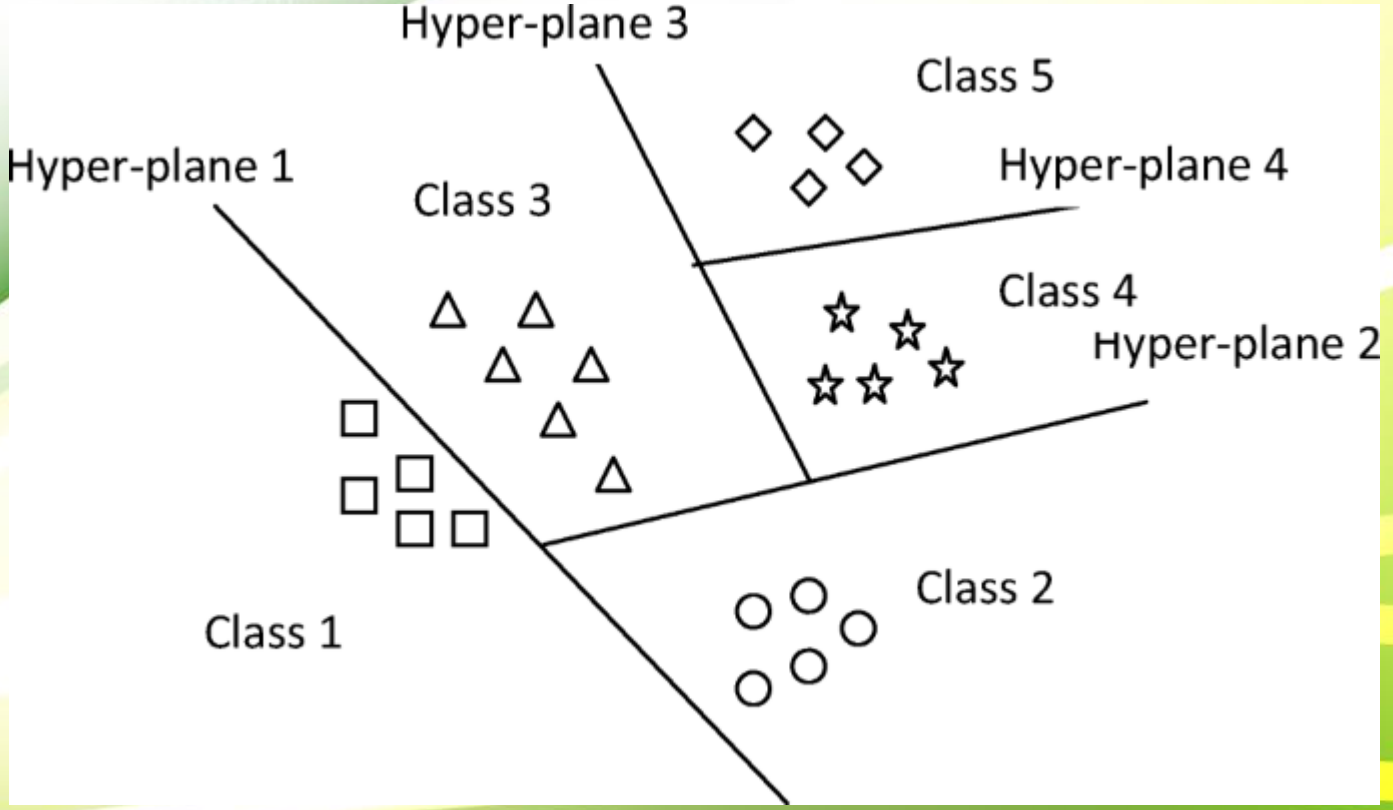


Support Vector Machine (SVM)

Two types of classification:

- **Binary classification** separates data points into two classes.
- **Multiclass classification** breaks down the dataset into multiple classes (uses hierarchical binary classification approach).

Multi-Class SVM

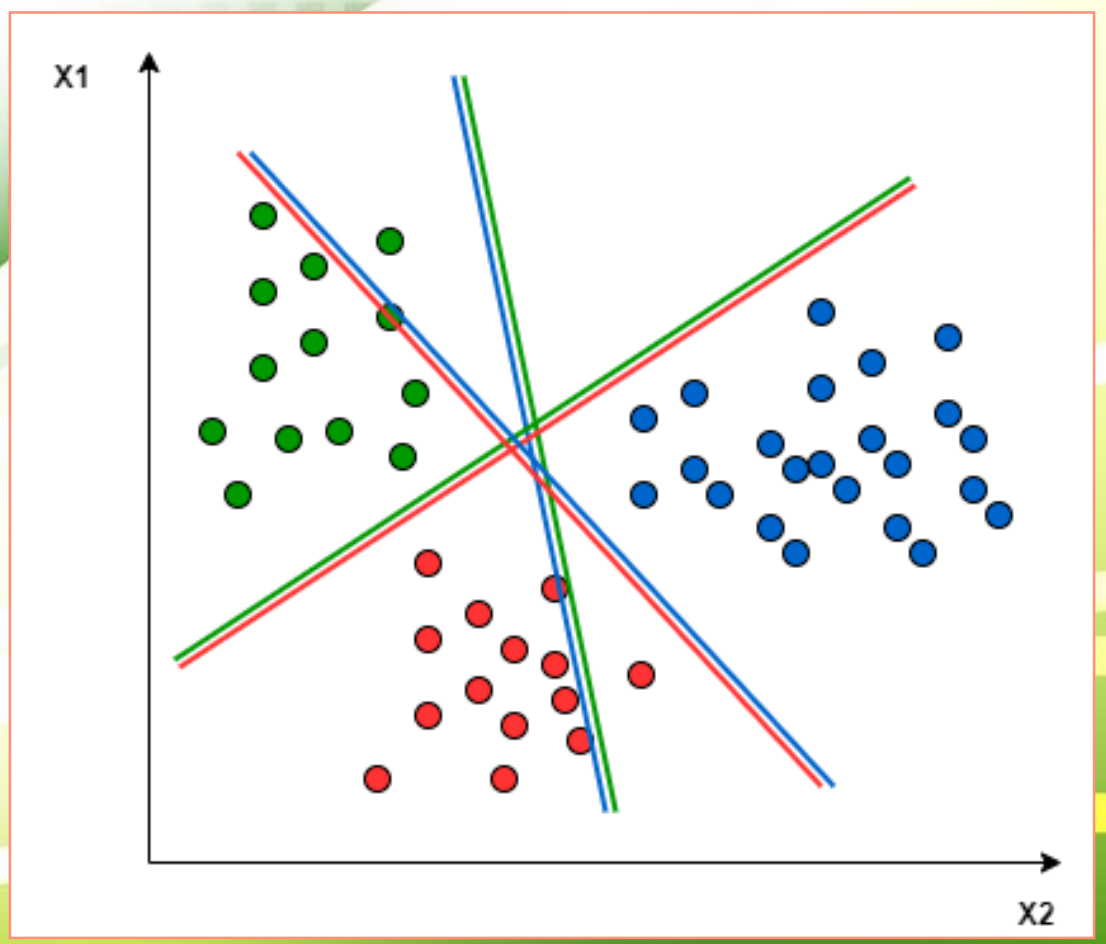


Support Vector Machine (SVM)

Two types of classification:

- **Binary classification** separates data points into two classes.
- **Multiclass classification** breaks down the dataset into multiple classes (uses hierarchical binary classification approach).

Multi-Class SVM



Support Vector Machine (SVM)

Pros:

- works really well with a clear margin of separation
- effective in high dimensional spaces.
- effective in cases where number of dimensions is greater than number of samples.
- Uses a subset of training points in decision function (called support vectors), so it is also memory efficient.

Cons:

- doesn't perform well with large data set; because the required training time is higher
- doesn't perform very well, when data set has more noise i.e. target classes are *overlapping*
- Doesn't directly provide probability estimates, these are calculated using expensive five-fold cross-validation.

Naive Bayes

- Supervised learning algorithm, used for solving **classification** problems.
- Based on **Bayes' Theorem**.
- Naive Bayes classifier assumes that presence of a feature in a class is not related to any other feature.
  - Every pair of features being classified is *(conditionally)* independent of each other.
- Classification algorithm for both binary and multi-class classification problems.
  - *Work with only categorical response variables and Large data sets.*
- **Naive**: Called Naïve because it assumes that occurrence of a certain feature is (conditionally) independent of occurrence of other features.
  - If fruit is identified based on color, shape, and taste; then red, spherical, and sweet fruit is recognized as apple.
  - Here each feature individually contributes to identify that it is an apple **without depending on each other.**
- **Bayes**: It is called Bayes because it depends on the principle of Bayes' Theorem.

Naive Bayes

Probability

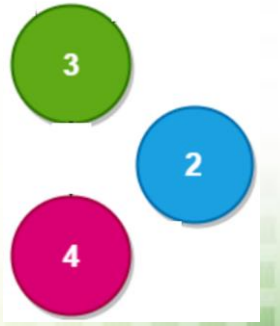
Events can be:

- **Independent** Each event is not affected by other events.
  - *Tossing a coin two times.*
  - *outcome of tossing coin for first time will not affect outcome of second event.*
- **Dependent** (Conditional) An event is affected by other events.
  - *Drawing 2 Cards from a Deck.*
  - *After taking one card from the deck there are fewer cards available, so the probabilities change.*
- **Mutually Exclusive** Two events can't happen at the same time.
  - *Outcomes of a single coin toss, which can result in either heads or tails, but not both.*
  - *Winning football & losing rugby played at same time in different venues.*

Naive Bayes

Probability

- Event A
- P(A)
- P(not A)



$$P(A) = \frac{\text{Number of favorable outcomes to A}}{\text{Total number of possible outcomes}}$$

P(A') = 1 - P(A). P(A')

Probability of Event A or Event B. **$P(A \cup B) = P(A) + P(B) - P(A \cap B)$**

Joint Probability = $P(A \cap B) = P(A) \times P(B)$

Naive Bayes

Probability

What is the probability of drawing a queen from a deck of cards?

$$P(A) = \frac{\text{Number of favorable outcomes to A}}{\text{Total number of possible outcomes}}$$

A deck of cards has 4 suits.

Each suit consists of 13 cards.

total number of possible outcomes = (4)(13) = 52

There can be 4 queens, one belonging to each suit.

number of favorable outcomes = 4.

card probability = 4 / 52 = 1 / 13

Naive Bayes

Probability

When two dice are rolled what is the probability of getting a sum of 8?

When two dice are rolled there are 36 possible outcomes.

To get the sum as 8 there are 5 favorable outcomes.

[(2, 6), (6, 2), (3, 5), (5, 3), (4, 4)]

$$P(A) = \frac{\text{Number of favorable outcomes to A}}{\text{Total number of possible outcomes}}$$

= 5 / 36

Naive Bayes

Probability

- **Marginal probability:** probability of an event irrespective of outcome of another variable
- **Joint probability P(A,B)**: probability that two events will both occur (likelihood of two events occurring together).
- **Conditional probability:** probability of one event occurring given that second event has happened.

$$P(X/Y) = \frac{P(X \cap Y)}{P(Y)}, \text{ if } P(Y) \neq 0$$

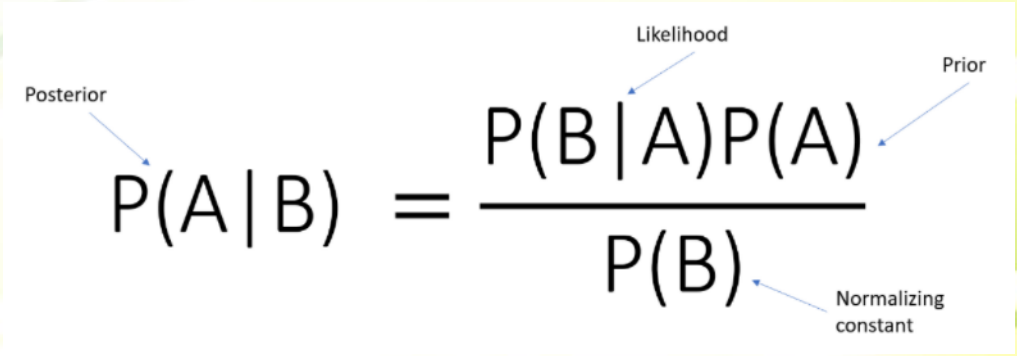
$$P(Y/X) = \frac{P(Y \cap X)}{P(X)}, \text{ if } P(X) \neq 0$$

Naive Bayes

Bayes' Theorem:

- Used to determine probability of a hypothesis with prior knowledge.
- It depends on conditional probability.
- Using Bayes theorem helps finding the probability of A, given that B occurred.

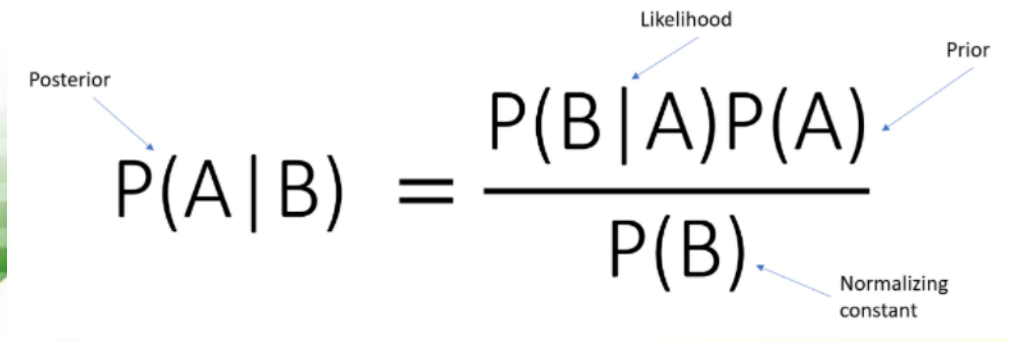
where,



- A and B are the events and P (B) ≠ 0
- **Conditional/Posterior probability: P(A/B), P(B/A)**
- **Marginal/prior probability: P(A), P(B)**
- P(A/B) is a **conditional probability** that describes the occurrence of event **A** is given that **B** is true.
- P(B/A) is a **conditional probability** that describes the occurrence of event **B** is given that **A** is true.
- P(X) and P(Y) are the probabilities of observing X and Y independently of each other.

Naive Bayes

Bayes' Theorem:



- A is the hypothesis and B is the evidence/condition.
- P(A) and P(B) is the independent probabilities of A and B.
- **P(A) is Prior Probability**: Probability of hypothesis before observing the evidence.
- **P(B) is Marginal Probability**: Probability of Evidence.
- **P(A|B) is Posterior probability**: Probability of hypothesis A on the observed event B.
- **P(B|A) is Likelihood probability**: Probability of the evidence given that the probability of a hypothesis is true.

Naive Bayes

Probability

Out of 10 people, 3 bought pencils, 5 bought notebooks and 2 got both pencils and notebooks. If a customer bought a notebook what is the probability that she also bought a pencil.

Using the concept of conditional probability in probability theory,

$$P(A \mid B) = P(A \cap B) / P(B)$$

A : event of people buying pencils

B : event people of buying notebooks.

$P(A) = 3 / 10 = 0.3$

$P(B) = 5 / 10 = 0.5$

$P(A \cap B) = 2 / 10 = 0.2$

$P(A \mid B) = 0.2 / 0.5 = 0.4$

Naive Bayes

- In **Bayesian interpretation**, probability determines "degree of belief."
- Bayes theorem connects degree of belief in a hypothesis before and after accounting for evidence.
  - *Example. If coin is tossed, either head or tail comes, and percent of occurrence of either is 50%.*
  - *If coin is flipped numbers of times, and outcomes are observed, the degree of belief may rise, fall, or remain same depending on outcomes.*

  - Proposition X and evidence Y,
  - P(X), the **prior**, is primary degree of belief in X
  - P(X/Y), the **posterior** is degree of belief having accounted for Y.
  - P(Y/X) / P(Y) represents the supports Y provides for X.

Naive Bayes

- **Chain rule of conditional probability:**

  $P(A,B) = p(A|B)\ p(B)$

  $P(A,B|C) = P(A|B,C)\ P(B|C)$

- Extend this for three variables:
  - *joint probability distribution of conditional probabilities.*

  $P(A,B,C) = P(A|\ B,C)\ P(B,C) = P(A|B,C)\ P(B|C)\ P(C)$

- General to n variables:

  $P(A1, A2, ..., An) = P(A1|\ A2, ..., An)\ P(A2|\ A3, ..., An)\ P(An\text{-}1|An)\ P(An)$

$$P(X/Y) = \frac{P(X \cap Y)}{P(Y)}, \text{ if } P(Y) \neq 0$$

$$P(Y/X) = \frac{P(Y \cap X)}{P(X)}, \text{ if } P(X) \neq 0$$

$$P(Y \cap X) = P(X \cap Y)$$

$$\text{or, } P(X \cap Y) = P(X/Y)P(Y) = P(Y/X)P(X)$$

$$\text{or, } P(X/Y) = \frac{P(Y/X)P(X)}{P(Y)}, \text{ if } P(Y) \neq 0$$

Naive Bayes

- Bayes theorem can be derived from the conditional probability:

$$P(X/Y) = \frac{P(X \cap Y)}{P(Y)}, \text{ if } P(Y) \neq 0$$

$$P(Y/X) = \frac{P(Y \cap X)}{P(X)}, \text{ if } P(X) \neq 0$$

- P (X∩Y) is **joint probability** of both X and Y being true.

$$P(Y \cap X) = P(X \cap Y)$$

$$\text{or, } P(X \cap Y) = P(X/Y)P(Y) = P(Y/X)P(X)$$

$$\text{or, } P(X/Y) = \frac{P(Y/X)P(X)}{P(Y)}, \text{ if } P(Y) \neq 0$$

- P(Y) = P(Y | X) * P(X) + P(Y | not X) * P(not X)
- **complement of X:** P(not X) = 1 – P(X)

Naive Bayes

Bag1 contains 4 white and 8 black balls and Bag2 contains 5 white and 3 black balls. From one of the bag one ball is drawn at random and the ball which is drawn comes out as black. Find the probability that the ball is drawn from Bag1.

Let E1, E2 and A be three events,
E1 = Event of selecting Bag1
E2 = Event of selecting Bag2
A = Event of drawing black ball

P(E1) = P(E2) = 1/2
P(drawing a black ball from Bag1) = P(A|E1) = 8/12 = 2/3
P(drawing a black ball from Bag2) = P(A|E2) = 3/8

P(A) = P(drawing a black ball) = P(A|E1) * P(E1) + P(A|E2) * P(E2)
= 2/3 * 1/2 + 3/8 * 1/2 = 25/48

According to Bayes' Theorem,
Probability(drawing a black ball from Bag1)
P(E1|A) = P(A|E1) * P(E1) / P(A)

According to Bayes' Theorem,
Probability(drawing a black ball from Bag1)
P(E1|A) = P(A|E1) * P(E1) / P(A)
= (2/3 * 1/2) / (25/48) = 16/25

Probability that ball is drawn from Bag1 is 16/25

Naive Bayes

Imagine you are a financial analyst at an investment bank. According to your research of publicly-traded companies, 60% of the companies that increased their share price by more than 5% in the last three years replaced their CEOs during the period.

At the same time, only 35% of the companies that did not increase their share price by more than 5% in the same period replaced their CEOs. Knowing that the probability that the stock prices grow by more than 5% is 4%, find the probability that the shares of a company that fires its CEO will increase by more than 5%.

Define the notation of probabilities.
P(I) –probability that stock price increases by 5% = (4%) = 0.04
P(R) –probability that CEO is replaced = (60%) = 0.6
P(R | I) –probability of CEO replacement given stock price has increased by 5% = (60%) = 0.6
P(I | R) –probability of stock price increases by 5% given that CEO has been replaced = ???

P(I`) = 1 – 0.4 = 0.6
P(R | I`) = (35%) = 0.35

According to Bayes' Theorem,
P(I | R) = P(R | I) * P(I) / P(R)

P(R) = P(R | I) * P(I) + P(R | I`) * P(I`)

$$P(A|B) = \frac{0.60 \times 0.04}{0.60 \times 0.04 + 0.35 \times (1 - 0.04)} = 0.067 \text{ or } 6.67\%$$

Naive Bayes

SpamAssassin works as a mail filter to identify spam in which users train the system. In emails, it considers patterns in words which are marked as spam by users. For Example, it may have learned that the word "release" is marked as spam in 30% of the emails. Concluding 0.8% of non-spam mails which includes the word "release" and 40% of all emails which are received by user is spam. Find probability that a mail is a spam if the word "release" seems in it.

P(R | S) = 0.30
P(R | N) = 0.008
P(S) = 0.40
=> P(N) = 0.60

P(S | R) = ?

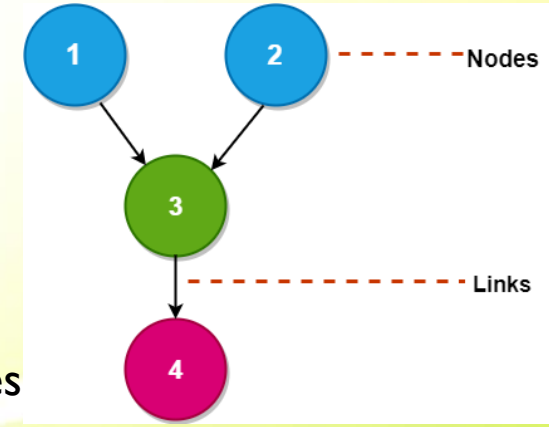
Now, using Bayes' Theorem:
P(S | R) = P(R | S) * P(S) / P(R)

P(R) = P(R | S) * P(S) + P(R | N) * P(N)
= 0.40 * 0.30 + 0.30 * 0.008 = 0.1224

Using Bayes' Theorem:
P(S | R) = P(R | S) * P(S) / P(R)
= 0.30 * 0.40 / 0.1224 = 0.980

Naive Bayes

- **Bayesian Network (Belief networks)** falls under classification of Probabilistic Graphical Modelling (PGM) that is utilized to compute uncertainties by utilizing the probability concept.
- Bayesian networks is used to show uncertainties through DAGs (**Directed Acyclic Graphs**).
- DAG gives a graphical model of the relationship.
- Contains a group of nodes and links.
- Every node represents a random variable; and edges define relationship between these variables
- Variables can be continuous or discrete values and may correspond to the actual attribute given to the data.
- DAG models uncertainty of an event taking place based on Conditional Probability Distribution (CDP) of each random variable.
- **Conditional Probability Table** (CPT) is used to represent CPD of each variable in network.

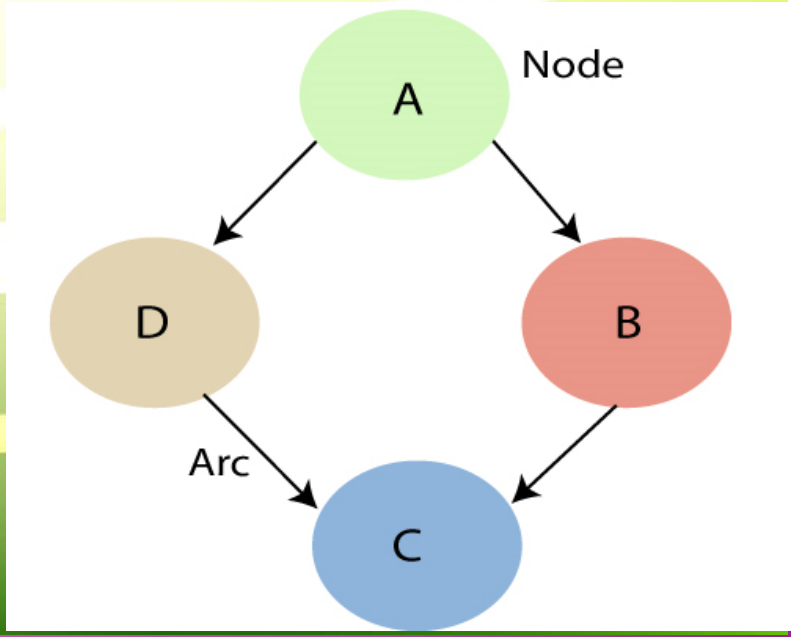


Train Strike	
Y	0.1
N	0.9

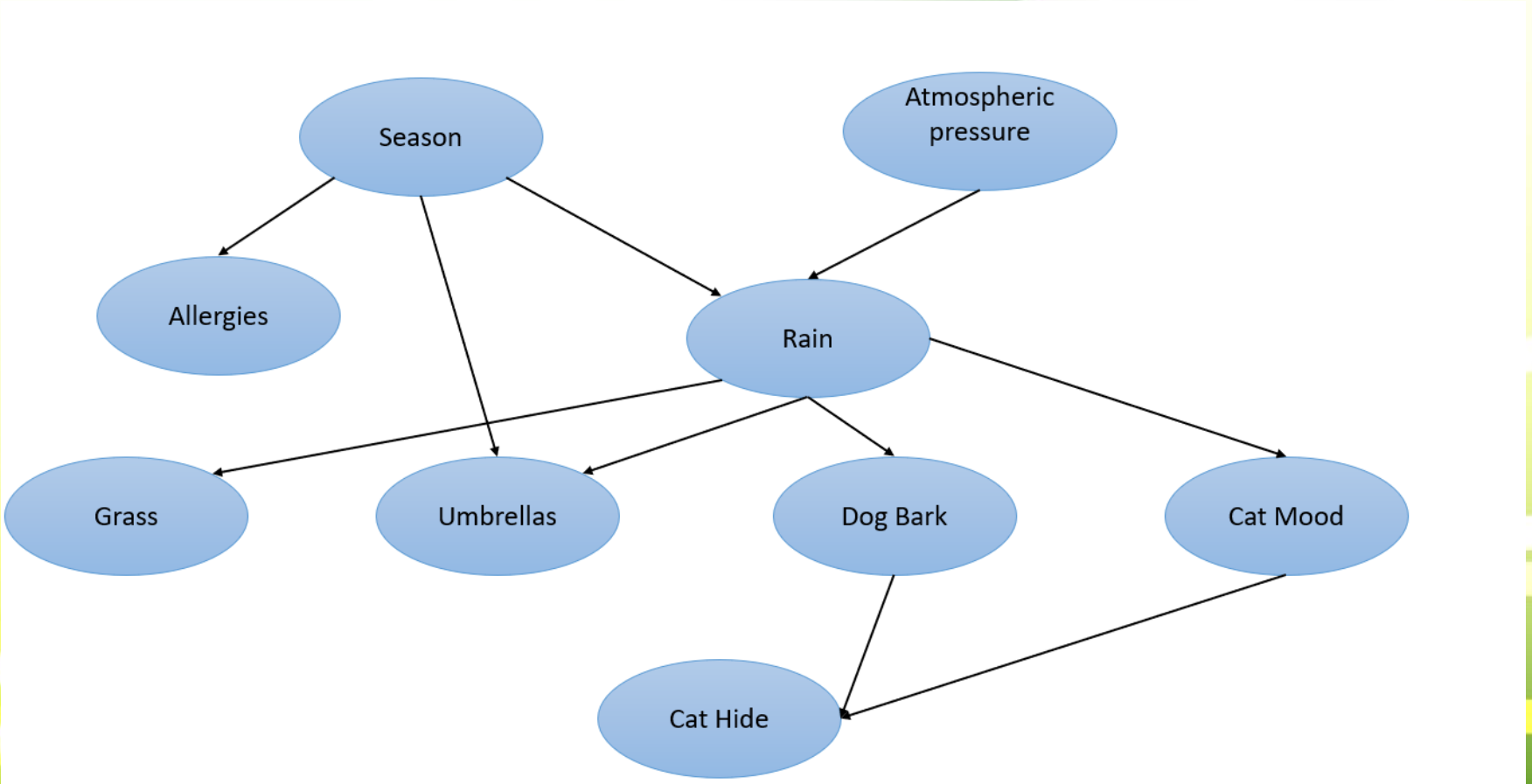
	Kelvin Late	
Train Strike	Y	N
Y	0.6	0.4
N	0.1	0.9

Naive Bayes

- Bayesian network graph is made up of nodes and Arcs (directed links).
- Each node corresponds to random variables (continuous or discrete).
- Arc or directed arrows represent causal relationship or conditional probabilities between random variables (connect pair of nodes in graph).
- These links represent that one node directly influence the other node, and if there is no directed link that means that nodes are independent with each other
  - In the diagram, A, B, C, and D are random variables represented by nodes of the network graph.
  - Node B is connected with node A by a directed arrow, then node A is called parent of Node B.
  - Node C is independent of node A.



Naive Bayes



An Example Bayesian Belief Network Representation

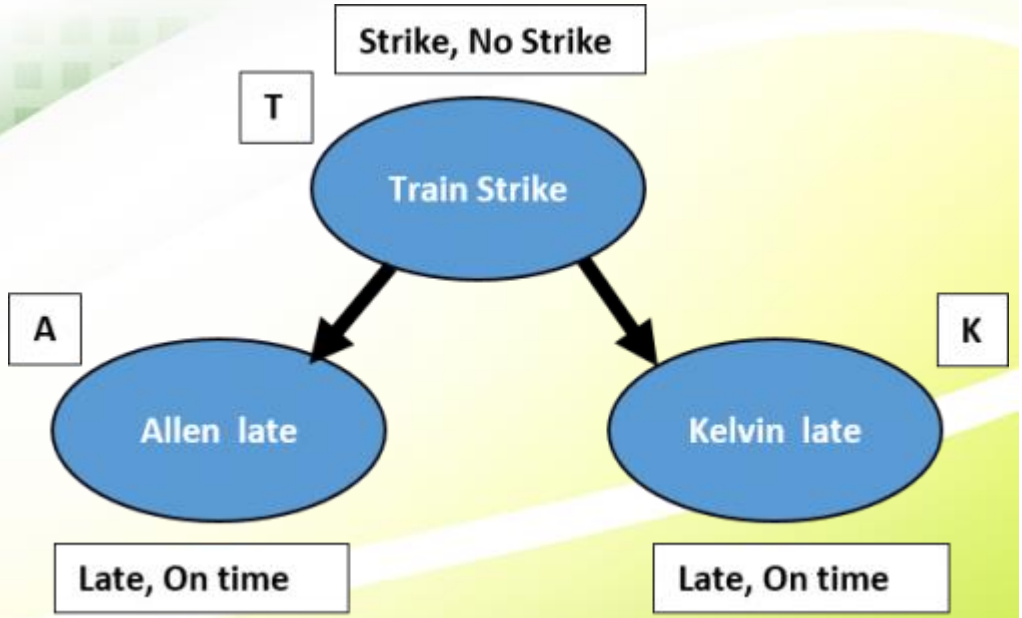
Naive Bayes

Example: Train strike influence on Allen's and Kelvin's work timings.

Problem: (1) Calculate the probability that allen will be on time. Calculate for kelvin also.

(2) Now, we came to know that Allen is late, but we do not know if there is a train strike. Find the probability.

(3) Can we know the probability that Kelvin will be late given we know that Allen is late?



Train Strike	
Y	0.1
N	0.9

	Kelvin Late	
Train Strike	Y	N
Y	0.6	0.4
N	0.1	0.9

	Allen Late	
Train Strike	Y	N
Y	0.7	0.3
N	0.6	0.4

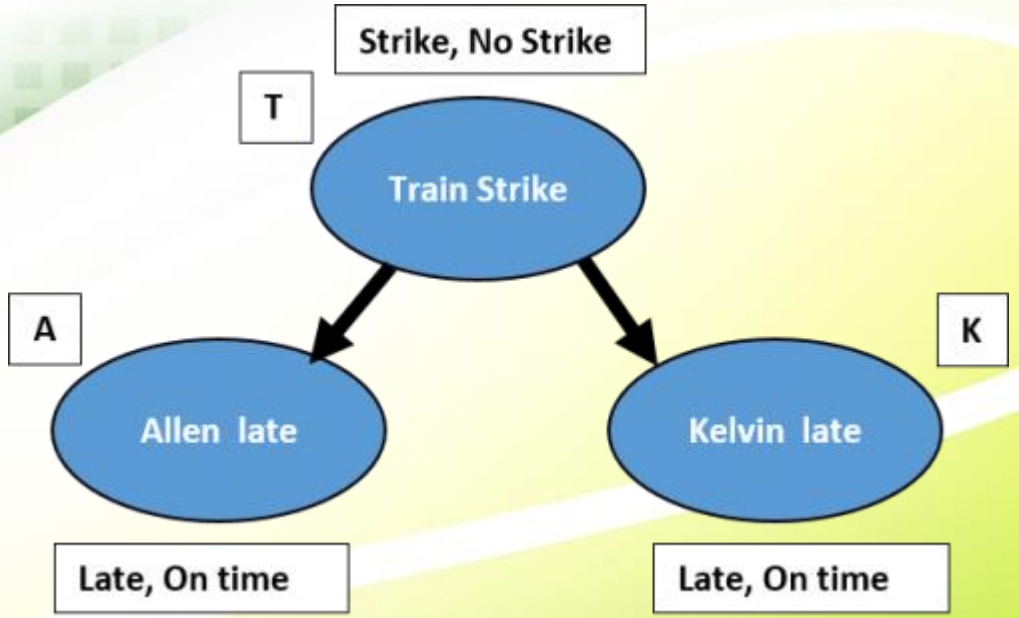
Naive Bayes

Example: Train strike influence on Allen's and Kelvin's work timings.

Problem: (1) Calculate the probability that allen will be late. Calculate for kelvin also.

(2) Now, we came to know that Allen is late, but we do not know if there is a train strike. Find the probability.

(3) Can we know the probability that Kelvin will be late given we know that Allen is late?



$P(A) = P(A|T)*P(T) + P(A|\sim T)*P(\sim T) == 0.7*0.1 + 0.6*0.9 = 0.61$

$P(K) = P(K|T)*P(T) + P(K|\sim T)*P(\sim T) = 0.6*0.1 + 0.1*0.9 = 0.15$

$P(T \mid A) = P(A \mid T) * P(T)/P(A) = 0.7*0.1/0.61 = 0.12$

$P(K) = P(K, T) + P(K, \sim T) = P(K|T)*P(T) + P(K|\sim T)*P(\sim T)$
$= 0.6*0.12 + 0.1*0.88 = 0.16$

Train Strike	
Y	0.1
N	0.9

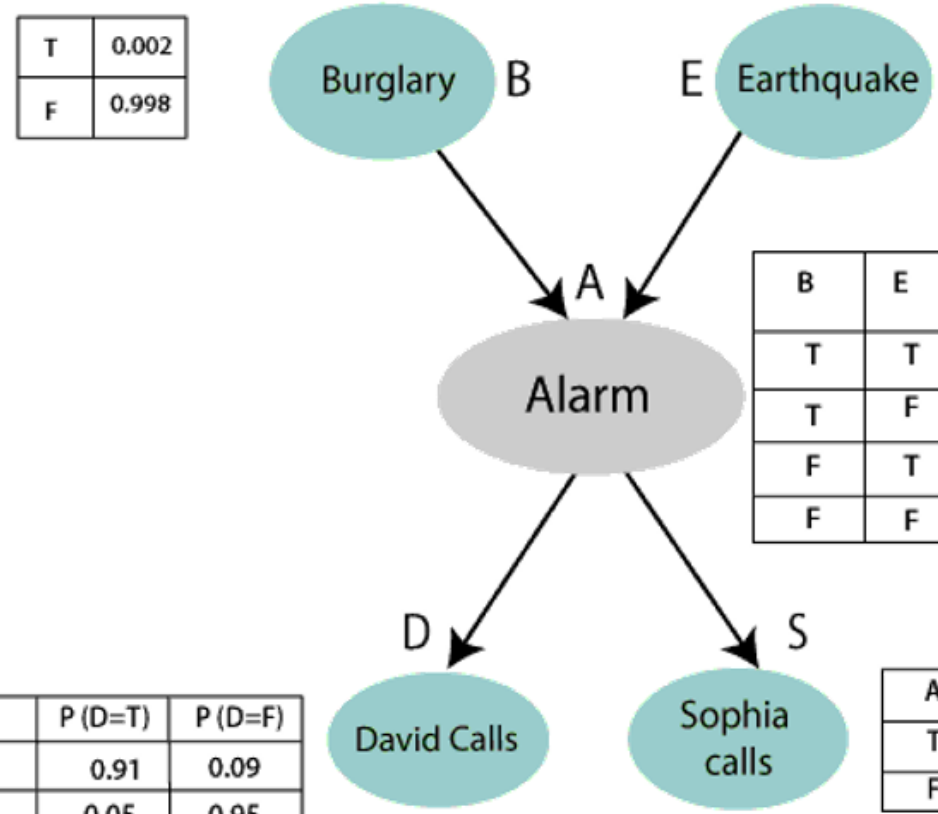
	Kelvin Late	
Train Strike	Y	N
Y	0.6	0.4
N	0.1	0.9

	Allen Late	
Train Strike	Y	N
Y	0.7	0.3
N	0.6	0.4

Naive Bayes

Example: Harry installed a new burglar alarm at his home to detect burglary. The alarm reliably responds at detecting a burglary but also responds for minor earthquakes. Harry has two neighbors David and Sophia, who have taken a responsibility to inform Harry at work when they hear the alarm. David always calls Harry when he hears the alarm, but sometimes he got confused with the phone ringing and calls at that time too. On the other hand, Sophia likes to listen to high music, so sometimes she misses to hear the alarm. Here we would like to compute the probability of Burglary Alarm.

Problem: Calculate the probability that alarm has sounded, but there is neither a burglary, nor an earthquake occurred, and David and Sophia both called the Harry.



T	0.002
F	0.998

T	0.001
F	0.999

B	E	P(A=T)	P(A=F)
T	T	0.94	0.06
T	F	0.95	0.04
F	T	0.69	0.69
F	F	0.001	0.999

A	P (D=T)	P (D=F)
T	0.91	0.09
F	0.05	0.95

A	P (S=T)	P (S=F)
T	0.75	0.25
F	0.02	0.98

Naive Bayes

- **List of all events occurring in this network:**
  - **Burglary (B)**
  - **Earthquake(E)**
  - **Alarm(A)**
  - **David Calls(D)**
  - **Sophia calls(S)**
- Events of problem statement in form of probability: **P[D, S, A, ~B, ~E]**
- Using joint probability distribution:

P[D, S, A, ~B, ~E] = P[D | S, A, ~B, ~E]. P[S, A, ~B, ~E]

=P[D | S, A, ~B, ~E]. P[S | A, ~B, ~E]. P[A, ~B, ~E]

= P [D| A]. P [S| A, ~B, ~E]. P[A, ~B, ~E]

= P[D | A]. P[S | A]. P[A| ~B, ~E]. P[~B, ~E]

= P[D | A]. P[S | A]. P[A| ~B, ~E]. P[~B |~E]. P[~E] **or**

= P[D | A]. P[S | A]. P[A| ~B, ~E]. P[~B]. P[~E]

- **Chain rule of conditional probability:**

$P(A,B) = p(A|B)\ p(B)$

$P(A,B|C) = P(A|B,C)\ P(B|C)$

- Extend this for three variables:
  - *joint probability distribution of conditional probabilities.*

$P(A,B,C) = P(A|\ B,C)\ P(B,C) = P(A|B,C)\ P(B|C)\ P(C)$

- General to n variables:

$P(A1, A2, ..., An) = P(A1|\ A2, ..., An)\ P(A2|\ A3, ..., An)\ P(An\text{-}1|An)\ P(An)$

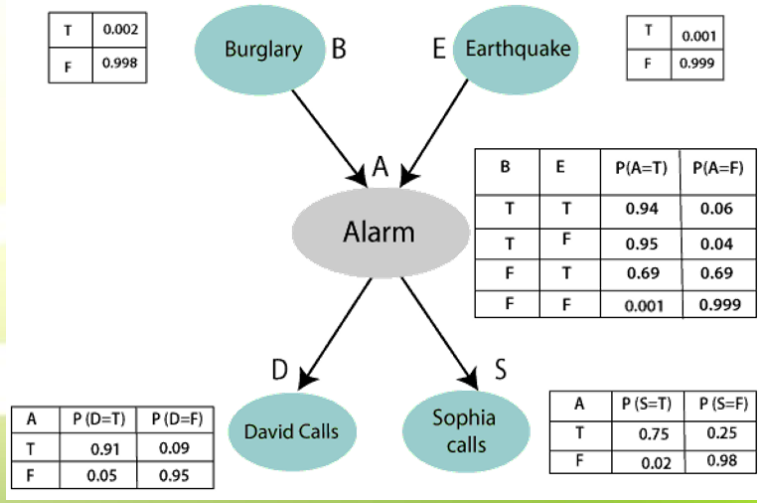
$$P(X/Y) = \frac{P(X \cap Y)}{P(Y)}, \text{ if } P(Y) \neq 0$$

$$P(Y/X) = \frac{P(Y \cap X)}{P(X)}, \text{ if } P(X) \neq 0$$

$P(Y \cap X) = P(X \cap Y)$

or, $P(X \cap Y) = P(X/Y)P(Y) = P(Y/X)P(X)$

or, $P(X/Y) = \frac{P(Y/X)P(X)}{P(Y)}$, if $P(Y) \neq 0$



T	0.002
F	0.998

T	0.001
F	0.999

B	E	P(A=T)	P(A=F)
T	T	0.94	0.06
T	F	0.95	0.04
F	T	0.69	0.69
F	F	0.001	0.999

A	P (D=T)	P (D=F)
T	0.91	0.09
F	0.05	0.95

A	P (S=T)	P (S=F)
T	0.75	0.25
F	0.02	0.98

joint probability distribution of conditional probabilities.

$P(A,B,C) = P(A|\ B,C)\ P(B,C) = P(A|B,C)\ P(B|C)\ P(C)$

Joint Probability = $P(A \cap B) = P(A) \times P(B)$

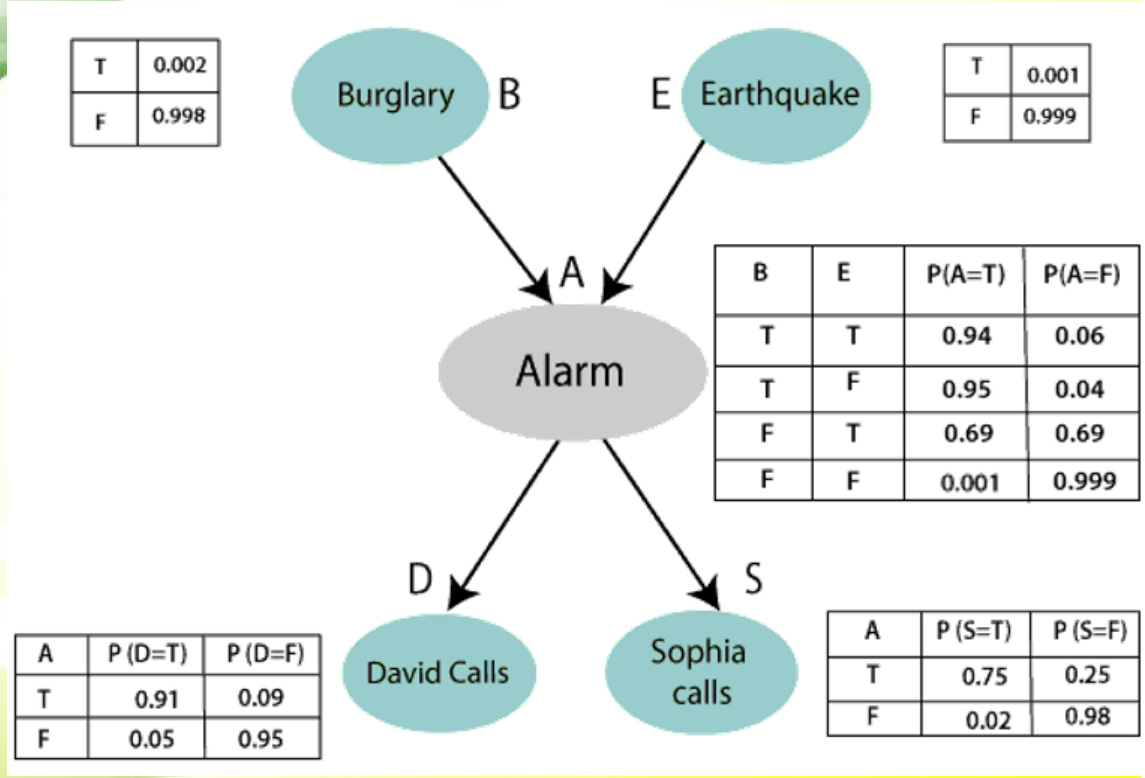
Naive Bayes

- **List of all events occurring in this network:**
  - **Burglary (B)**
  - **Earthquake(E)**
  - **Alarm(A)**
  - **David Calls(D)**
  - **Sophia calls(S)**
- Events of problem statement in form of probability: **P[D, S, A, ~B, ~E]**
- Using joint probability distribution:

P[D, S, A, ~B, ~E] = P[D | A]. P[S | A]. P[A| ~B, ~E]. P[~B]. P[~E]

= 0.91* 0.75* 0.001* 0.998*0.999

= 0.00068045



T	0.002
F	0.998

T	0.001
F	0.999

B	E	P(A=T)	P(A=F)
T	T	0.94	0.06
T	F	0.95	0.04
F	T	0.69	0.69
F	F	0.001	0.999

A	P (D=T)	P (D=F)
T	0.91	0.09
F	0.05	0.95

A	P (S=T)	P (S=F)
T	0.75	0.25
F	0.02	0.98

Naive Bayes

- Supervised learning algorithm, used for solving **classification** problems.
- Naive Bayes classifier assumes that presence of a feature in a class is not related to any other feature (conditionally independent).
- Naive Bayes classifiers are a collection of classification algorithms based on **Bayes' Theorem**.
  - It is not a single algorithm but a family of algorithms where all of them share a common principle.
    - **Gaussian Naïve Bayes**
    - **Multinomial Naïve Bayes**
    - **Bernoulli Naïve Bayes**

Naive Bayes

- Consider the problem of playing golf.
- Build a predictive model whether the day is suitable for playing golf, given the features of the day

	Outlook	Temperature	Humidity	Windy	Play Golf
0	Rainy	Hot	High	False	No
1	Rainy	Hot	High	True	No
2	Overcast	Hot	High	False	Yes
3	Sunny	Mild	High	False	Yes
4	Sunny	Cool	Normal	False	Yes
5	Sunny	Cool	High	True	No
6	Overcast	Cool	Normal	True	Yes
7	Rainy	Mild	High	False	Yes
8	Rainy	Cool	Normal	False	Yes
9	Sunny	Mild	Normal	False	No
10	Rainy	Mild	Normal	True	Yes
11	Overcast	Mild	Normal	True	Yes
12	Overcast	Hot	Normal	False	Yes
13	Sunny	Mild	High	True	No

Naive Bayes

	Outlook	Temperature	Humidity	Windy	Play Golf
0	Rainy	Hot	High	False	No
1	Rainy	Hot	High	True	No
2	Overcast	Hot	High	False	Yes
3	Sunny	Mild	High	False	Yes
4	Sunny	Cool	Normal	False	Yes
5	Sunny	Cool	High	True	No
6	Overcast	Cool	Normal	True	Yes
7	Rainy	Mild	High	False	Yes
8	Rainy	Cool	Normal	False	Yes
9	Sunny	Mild	Normal	False	No
10	Rainy	Mild	Normal	True	Yes
11	Overcast	Mild	Normal	True	Yes
12	Overcast	Hot	Normal	False	Yes
13	Sunny	Mild	High	True	No

- Dataset is divided into two parts, **feature matrix** and the **response vector**.
- Feature matrix contains all **dependent features** 'Outlook', 'Temperature', 'Humidity' and 'Windy'.
- Response vector contains value of **class variable** (output) 'Play golf' for each row of feature matrix.
- Naive Bayes goes with assumption that each feature makes an **independent** and **equal** contribution to outcome.
  - i.e. No pair of features are dependent.
    - Temperature being 'Hot' has nothing to do with humidity or outlook being 'Rainy' has no effect on winds.
  - Secondly, each feature is given same weight (importance).
    - Knowing only temperature and humidity alone can't predict outcome **accurately**.
    - None of the attributes is irrelevant and assumed to be contributing equally to the outcome.
  - *Assumptions made by Naive Bayes are not generally correct in real-world situations, but in theoretical concepts only.*

Naive Bayes

Bayes theorem can be rewritten as:

$$P(y|X) = \frac{P(X|y)P(y)}{P(X)}$$

Variable **y** is output/response (play golf) → whether suitable to play or not given the conditions.

Variable **X** represent parameters/features.

$$X = (x_1, x_2, x_3, ..., x_n)$$

x_1,x_2….x_n represent features → outlook, temperature, humidity, windy.

By substituting for X and expanding using Chain rule;

$$P(y|x_1, ..., x_n) = \frac{P(x_1|y)P(x_2|y)...P(x_n|y)P(y)}{P(x_1)P(x_2)...P(x_n)}$$

	Outlook	Temperature	Humidity	Windy	Play Golf
0	Rainy	Hot	High	False	No
1	Rainy	Hot	High	True	No
2	Overcast	Hot	High	False	Yes
3	Sunny	Mild	High	False	Yes
4	Sunny	Cool	Normal	False	Yes
5	Sunny	Cool	High	True	No
6	Overcast	Cool	Normal	True	Yes
7	Rainy	Mild	High	False	Yes
8	Rainy	Cool	Normal	False	Yes
9	Sunny	Mild	Normal	False	No
10	Rainy	Mild	Normal	True	Yes
11	Overcast	Mild	Normal	True	Yes
12	Overcast	Hot	Normal	False	Yes
13	Sunny	Mild	High	True	No

Naive Bayes

- Consider the problem of playing golf.
- Build a predictive model whether the day is suitable for playing golf, given the weather is "Sunny".

P(Y|S) = P(S|Y)*P(Y)/P(S)
P(~Y|S) = P(S|~Y)*P(~Y)/P(S)

P(Y|S) **P(~Y|S)**

Frequency table for Weather Conditions:

Weather	Yes	No
Overcast	5	0
Rainy	2	2
Sunny	3	2
Total	10	5

Likelihood table weather condition:

Weather	No	Yes	
Overcast	0	5	5/14= 0.35
Rainy	2	2	4/14=0.29
Sunny	2	3	5/14=0.35
All	4/14=0.29	10/14=0.71	

Applying Bayes'theorem:

P(Yes|Sunny)= P(Sunny|Yes)*P(Yes)/P(Sunny) = 0.3*0.71/0.35= **0.60**

P(Sunny|Yes)= 3/10= 0.3 P(Sunny)= 0.35 P(Yes)=0.71

P(No|Sunny)= P(Sunny|No)*P(No)/P(Sunny) = 0.5*0.29/0.35 = **0.41**

P(Sunny|NO)= 2/4=0.5 P(Sunny)= 0.35 P(No)= 0.29

P(Yes|Sunny) > P(No|Sunny) → Hence on a Sunny day, Player can play game.

	Outlook	Play
0	Rainy	Yes
1	Sunny	Yes
2	Overcast	Yes
3	Overcast	Yes
4	Sunny	No
5	Rainy	Yes
6	Sunny	Yes
7	Overcast	Yes
8	Rainy	No
9	Sunny	No
10	Sunny	Yes
11	Rainy	No
12	Overcast	Yes
13	Overcast	Yes

Naive Bayes

- Consider the problem of playing golf. Build a predictive model whether the day is suitable for playing (y), given conditions **X = "Sunny", "Hot", "Normal", "False"**

P(Y|X) & P(~Y|X) .... P(Y|X) = P(Y|S,H,N,F)

	Outlook	Temperature	Humidity	Windy	Play Golf
0	Rainy	Hot	High	False	No
1	Rainy	Hot	High	True	No
2	Overcast	Hot	High	False	Yes
3	Sunny	Mild	High	False	Yes
4	Sunny	Cool	Normal	False	Yes
5	Sunny	Cool	High	True	No
6	Overcast	Cool	Normal	True	Yes
7	Rainy	Mild	High	False	Yes
8	Rainy	Cool	Normal	False	Yes
9	Sunny	Mild	Normal	False	No
10	Rainy	Mild	Normal	True	Yes
11	Overcast	Mild	Normal	True	Yes
12	Overcast	Hot	Normal	False	Yes
13	Sunny	Mild	High	True	No

Outlook

	Yes	No	P(yes)	P(no)
Sunny	2	3	2/9	3/5
Overcast	4	0	4/9	0/5
Rainy	3	2	3/9	2/5
Total	9	5	100%	100%

Temperature

	Yes	No	P(yes)	P(no)
Hot	2	2	2/9	2/5
Mild	4	2	4/9	2/5
Cool	3	1	3/9	1/5
Total	9	5	100%	100%

Humidity

	Yes	No	P(yes)	P(no)
High	3	4	3/9	4/5
Normal	6	1	6/9	1/5
Total	9	5	100%	100%

Wind

	Yes	No	P(yes)	P(no)
False	6	2	6/9	2/5
True	3	3	3/9	3/5
Total	9	5	100%	100%

Play		P(Yes)/P(No)
Yes	9	9/14
No	5	5/14
Total	14	100%

Naive Bayes

- Consider the problem of playing golf. Build a predictive model whether the day is suitable for playing (y), given conditions **X = "Sunny", "Hot", "Normal", "False"**

P(Y|X) = P(Y|S,H,N,F)

$$P(y|x_1,\ldots,x_n) = \frac{P(x_1|y)P(x_2|y)\ldots P(x_n|y)P(y)}{P(x_1)P(x_2)\ldots P(x_n)}$$

P(Y|X) = P(S|Y)P(H|Y)P(N|Y)P(F|Y)P(Y) / P(X)

Joint Probability = P(A ∩ B) = P(A) x P(B)

P(~Y|X) = P(S|~Y)P(H|~Y)P(N|~Y)P(F|~Y)P(~Y) / P(X)

** P(X) is common in both → ignore P(X)*

P(Y|X) = (2/9)*(2/9)*(6/9)*(6/9)*(9/14) = 0.0141

P(~Y|X) = (3/5)*(2/5)*(1/5)*(3/5)*(5/14) = 0.0068

Normalization: P(Y|X) = 0.0141/(0.0141+0.0068) = 0.67

P(~Y|X) = 0.0068/(0.0141+0.0068) = 0.33

P(Y|X) > P(~Y|X) → Game should be played today ('Yes')

Outlook

	Yes	No	P(yes)	P(no)
Sunny	2	3	2/9	3/5
Overcast	4	0	4/9	0/5
Rainy	3	2	3/9	2/5
Total	9	5	100%	100%

Temperature

	Yes	No	P(yes)	P(no)
Hot	2	2	2/9	2/5
Mild	4	2	4/9	2/5
Cool	3	1	3/9	1/5
Total	9	5	100%	100%

Humidity

	Yes	No	P(yes)	P(no)
High	3	4	3/9	4/5
Normal	6	1	6/9	1/5
Total	9	5	100%	100%

Wind

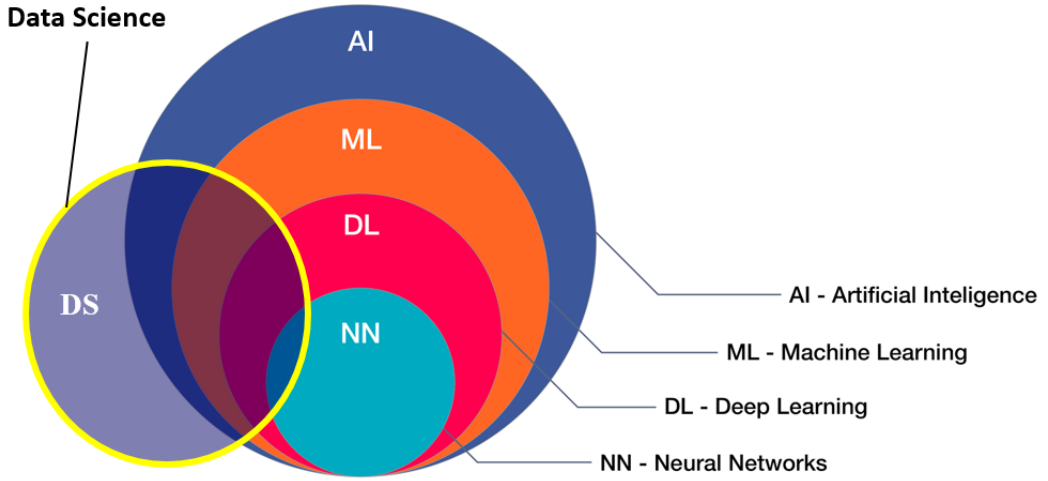
	Yes	No	P(yes)	P(no)
False	6	2	6/9	2/5
True	3	3	3/9	3/5
Total	9	5	100%	100%

Play		P(Yes)/P(No)
Yes	9	9/14
No	5	5/14
Total	14	100%

DEEP LEARNING

- Inspired by the information processing patterns found in human brain.
- Brain tries to decipher the information it receives; through labelling and assigning the items into various categories.
- Whenever we receive new information, brain tries to compare it to a known item before making sense of it → DL applies similar logic through DL algorithms.
- Neural Network (NN), artificial neural networks (ANNs) are types of techniques to imitate the way our brains make decisions.

DEEP LEARNING



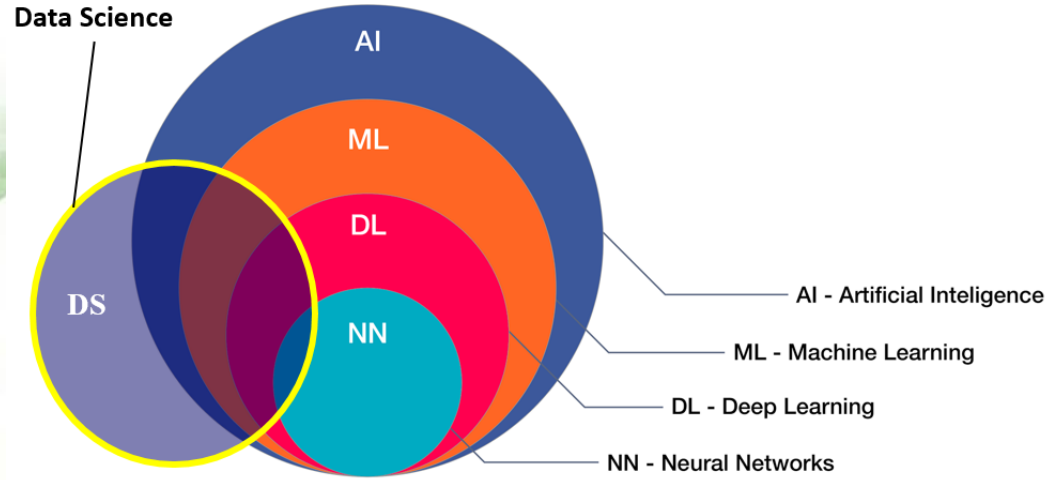
DL is extensively used for complex problems:

- Google search optimization,
- Tensorflow applications,
- Neural network applications,
- Self driving car,
- Image processing applications,
- Virtual assistance (Alexa, Google assistance),
- User recommendation on e-commerce/digital content websites *(Netflix, amazon etc),*
- *NLP applications; etc.*

ML vs. DL

	Machine Learning	Deep Learning
Data Dependencies	Superior performance on a small and medium dataset	Performs excellent on a big dataset
Hardware dependencies	Performs on a low-end machine	Preferable requires a machine with GPU. Deep Learning performs on a noteworthy matrix multiplication
Feature engineering	Carefully understand the features of how it represents the data	Required to understand the specific best functionality that represents the data
Execution time	From a few minutes to hours	It requires a time of up to 2-3 weeks.
Interpretability	Some algorithms are easy to interpret like, logistic and decision tree. Whereas some are almost impossible like, SVM and XGBoost	Difficult to impossible

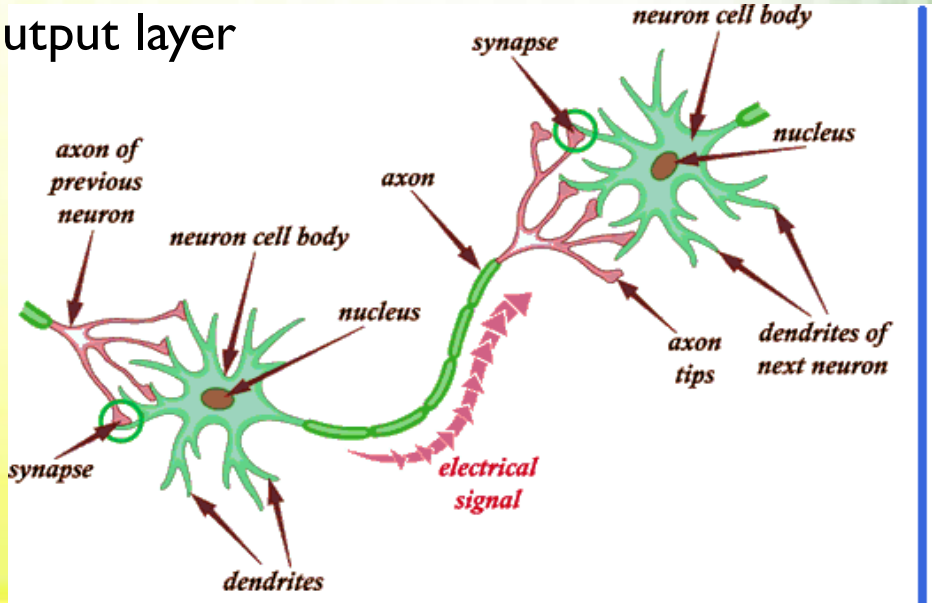
NEURAL NETWORK



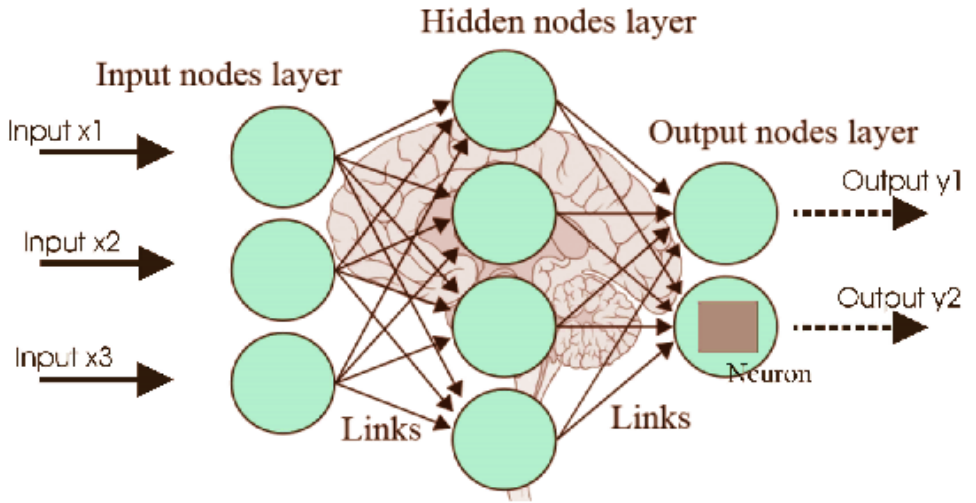
- NN enables ML and DL.
- Conceptual existence for over a centaury.
- ML/DL uses the NN algorithm to do their work.
- NN is specific group of algorithms used for ML that models the data using graphs of Artificial Neurons.
- Neurons are mathematical models that *"mimics approximately how a neuron in human brain works"*.
- Series of algorithms that help to recognize underlying relationships in a set of data through a process that mimics the way human brain operates.

NEURAL NETWORK

- Input Layer
- Hidden layer
- Output layer



Neuron in our brain.



Neural Network develop using A.I.

NEURAL NETWORK

- "Neuron" in neural network is a mathematical function that collects and classifies information according to a specific architecture.
- NN is a strong statistical tool for classification and regression analysis.
- NN contains layers of interconnected nodes.
- Each node is a perceptron and is similar to a multiple linear regression.
- Perceptron feeds the signal produced by a multiple linear regression into an activation function that may be nonlinear.

END