Financial Analytics:

(1) Decision Trees :- → classification Tree
→ Regression Tree

i) A decision tree is a supervised machine learning algorithm used for classification and regression tasks

ii) It works by recursively splitting the data into subsets based on the value of input features and assigning values to each subset based on the majority class / target variable

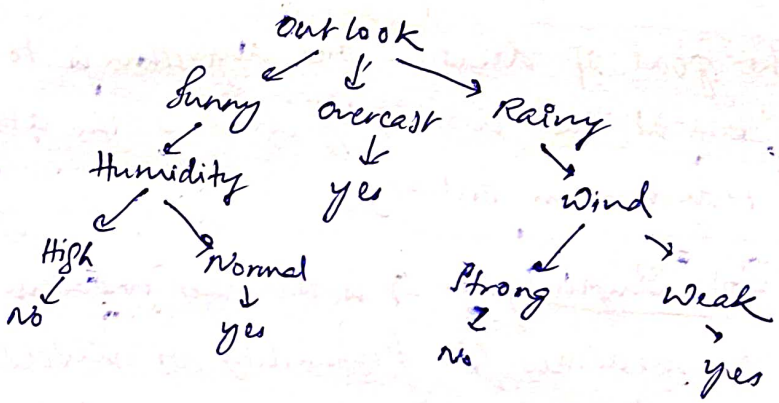
iii) Or The resultant hierarchical tree of decisions, each node represents a feature, each branch represents a possible value of that feature and each leaf node represents a final decision/outcome

iv) Decision trees are most suitable where the input features have discrete/categorical values, but they can also work with continuous vars.

v) They are particularly useful for problems involving large number of features as it helps in reducing ~~overfit~~ overfitting and improves generalisation

Problem use cases :
1) customer segmentation
2) Medical diagnosis
3) Sentiment Analysis
4) Build ML Techniques like Random Forest / Boosting

Example : Play Tennis :



PTO

Determination of quality of a split at each node of tree

① **Entropy** → It is the measure of impurity / randomness in a set of data. In decision trees, it is used to measure the impurity of a set of values wrt their class labels

If a set has all the values belonging to the same class, then entropy is 0. If the converse is true, then entropy is 1. (i.e. equal values of each class in the set)

$$\text{Entropy}(S) = -\sum p(i) \log_2 p(i)$$

$S \rightarrow$ Set of values ; $p(i) \rightarrow$ proportion of values in set S which belongs to class (i)

② **Information Gain** → It is the measure of the reduction in entropy achieved by a split.

It is calculated as the difference between the entropy of parent node and weighted avg of child nodes resulting from the split.

$$\text{Info gain}(S, A) = \text{Entropy}(S) - \sum \left(\frac{|S_v|}{|S|}\right) \times \text{Entropy}(S_v)$$

$A \rightarrow$ attribute being used for split, $S_v \rightarrow$ Set of values at child node

The goal of decision tree algorithm is to maximize the info gain at each node, by choosing the attribute that results in largest reduction in entropy

③ **Gini Impurity** → It is another measure of impurity/randomness in a split. It measures the probability of misclassifying a randomly chosen example in a set, if it were randomly labeled according to class distribution

$$\text{Gini}(S) = 1 - \sum p(i)^2$$

It ranges from 0 to 1 → complete impurity
0 → complete purity

Gini is preferred over Entropy because it is computationally less expensive, and can handle categorical/continuous variables more effectively

~~Decision Trees~~

Advantages

(i) Can handle both categorical/continuous variables

(ii) Can handle missing values/large data

(iii) Can perform feature selection, (reduce ~~avoid~~ complexity/improve accuracy)

Disadvantages

(i) Prone to overfitting

(ii) Can't capture interaction between variables

(iii) Limited to hierarchical data

---

(2) Support vector machines :-

i) It is a machine learning algorithm that can be used for classification and regression purpose

ii) The key idea behind SVM is to find a hyperplane that separates the data into classes and attempts to maximise the margin between the classes

iii) The margin is defined as the distance between the hyperplane and closest data points from each class, and maximisation of the margin ensures the best generalisation of the model.

iv) Kernelization :-

Not all data can be separated by a linear hyperplane. In such cases, SVMs use kernelization to transform the data into a higher dimensional space, where the classes might be linearly seperable

v) The role of kernel is to compute the dot product between the transformed data points. This is known as the Kernel Trick.

It is computationally efficient, as it avoids the need to compute the high-dimensional feature space explicitly.

Eg. linear, polynomial, RBF (radial based function), sigmoid

[choice depends on specific problem-at-hand and ~~the~~ data characteristics]

Advantages :

(i) Effective in high dimensional spaces

(ii) Can handle non-linear/unbalanced data

(iii) Robust to overfitting

Disadvantages :-

(i) computationally expensive

(ii) difficult to interpret / visualize

(iii) Sensitive to choice of kernel

(3) K-Neighbors :-

i) It is a machine learning algorithm used for classification and regression purposes

ii) The KNN consists of two steps - Training and Prediction.

iii) During training, the KNN algo simply stores the training data

iv) During prediction, the algo takes a new, unseen data point as input and finds the K closest data points in the training set.

The distance is calculated using Euclidean / Manhattan distance. The K- nearest points are then used to make prediction ~~on~~ for the new data point

v) For classification, the most common label is ~~as~~ assigned and for regression, the avg/median value ~~of~~ is used as predicted value

vi) The value of K can be chosen based on specic problem and data characteristics.

Larger K leads to smoother boundaries but also increases bias, while Smaller K " " complex boundaries but comes at a cost of ~~in~~ higher variance, and also increases risk of overfitting.

Advantages :-

(1) Non-parametric - no assumptions about the data underlying
(2) Robust to noisy data
(3) No training required
(4)* versatile - both classification and regression
(5)* Simple and easy to understand

Disadvantages

(1) computationally expensive
(2) Sensitive to distance metric
(3) performance $\propto \frac{1}{\text{dimensionality}}$
(4) Might be biased towards majority class in the case of imbalanced data
(5) Requires careful normalization, as features with large ranges dominate the distance metric

(4) Clustering Algorithms

(i) Partitional clustering :- (Ex K-means, k-medoids)

→ It is a clustering algorithm that aims to partition a dataset into a pre-determined number of clusters.

→ The number of clusters is specified by the user and the algo iteratively assigns data points to the closest cluster center based on their distance to the centroid

→ Useful to image segmentation, customer segmentation, anomaly detection

→ Disadvantage → Sensitive to initial selection of centroids

(ii) Hierarchical clustering

→ It is a clustering algorithm that creates a hierarchy of clusters. The clusters are created by → merging small cluster into larger (agglomerative)
→ recursively dividing larger into smaller clusters (divisive)

→ Agglomerative clustering → Each data point starts as its own cluster, and the algo iteratively merges the two closest clusters until one cluster remains. This results is a dendrogram that shows the hierarchy

→ Divisive clustering → All data points starts in a single cluster. The algo recursively divides the cluster into smaller clusters until each subcluster contains only a data point. This results in a dendrogram that shows the hierarchy of clusters.

→ Advantages → Doesn't require the user to input number of clusters. instead they can look at the dendrogram and determine where to cut the tree to obtain desired number of clusters.

Disadvantages → computationally expensive for large datasets
dependent on choice of distance metric and linkage method.

(iii) Intensity-based clustering :- (Mean shift Algorithm)

→ It is a clustering algorithm that groups data points based on their intensity in a high-dimensional space

→ The algo measures the prominity of data points based on their intensity values rather their distance from each other

→ The algo starts by defining a density measure based on the intensity of the data.

→ Then, it identifies regions in the data space where density exceeds a certain threshold (density parameter). These regions are considered as potential cluster centers

→ The algo expands each cluster by incorporating all neighboring data points that have density above the density parameter. until all data points are assigned to a cluster.

→ Disadvantage → performance is sensitive to choice of density parameter and kernel function used to estimate the density

---

Difference between Hierarchical and Partitional clustering

Partitional	Hierarchical
(i) Faster run time due to pre-set number of clusters	(i) Slower than partitional clustering
(ii) Requires stronger initial assumptions - number of clusters, initial centers	(ii) Requires only a similarity measure / distance metric
(iii) Requires the user to input no. number of clusters(k) to start working	(iii) Doesn't require any input parameters to start running
(iv) Returns exactly k clusters	(iv) Returns a subjective division of clusters in the form of a hierarchical dendrogram
(v) Computationally cheap	(v) computationally expensive

Neural Network - Single Layer perceptron :

→ It is the simplest form of neural network that consists of a single layer of neurons

→ The single layer perceptron takes input data, processes it through a linear combination of weights and outputs a prediction/classification.

→ The output of the perceptron is determined by a weighted sum of the input features, which is then passed through an activation function that maps the output to a binary value.

→ Mathematically, output (y) = $f(w_1 x_1 + w_2 x_2 + \dots + w_n x_n + b)$

$x_k \rightarrow$ input features ; $w_k \rightarrow$ weights ; $b \rightarrow$ bias ; $f(\cdot) \rightarrow$ activation function

→ The activation function is used to introduce non-linearity to the output of the perceptron. Eg. Step function

Neural Network - Multi Layer perceptron :

→ MLP (Multi-layer perceptron) is a type of neural network that consists of multiple layers of neurons.

→ It consists of an input layer, one or more hidden layers and output layers. The input layer receives the input data, ~~and each neuron in the~~ in the hidden layers performs the computations on the input data and the output layer generates the final output of the network.

→ Each ~~neu~~ neuron in an MLP has its own set of weights, which are learned during training.

→ The activation function used in each neuron plays an important role in the performance of the MLP. Eg. Sigmoid function, RELU (Rectified linear unit)

→ It is used for a variety of tasks - classification, regression and pattern recognition. Application in fields like computer vision, NLP and Speech recognition.

→ Disadvantage → prone to severe overfitting if model is too complex

Back propagation Algorithm :

→ It is used for training artificial neural networks. It is a supervised algorithm that works by iteratively adjusting the weights of the neural network to minimize the error between the predicted output and the true output.

→ It consists of two phases - forward pass, backward pass

→ Forward pass → • Input data is fed into the neural network
- Output is computed by propagating the input forward through the layers of the network.
- At each layer, the input is transformed by multiplying it with the neurons and adding a bias term, then passing the result through the activation function.

→ Backward pass →
- The error between the predicted output and the true output is computed using a loss function (like MSE, cross-entropy loss)
- The error is propagated backwards through the layers.
- The gradient of the error ~~with respect~~ wrt each weight and bias term is computed
- After gradients are calculated, the weights and biases are updated using an optimization algorithm, scaled by a learning rate.
- The process is repeated for multiple epochs until the error on the training data is minimized.

→ Backpropagation is an efficient way to train neural networks, but it can be prone to getting stuck in local optima where gradient descent is very small.

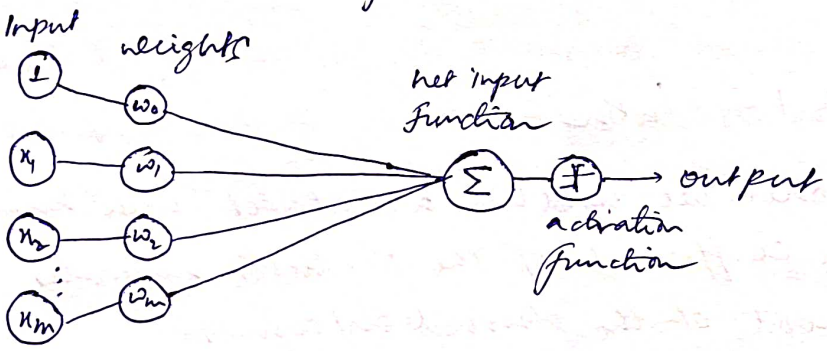
To address this, techniques like momentum, ~~adapta~~ adaptive learning rates, regularisation can be used to improve the convergence of the algorithm and prevent overfitting

Q. Explain artificial neural networks based on perception concept with diagram

1) ANNs are computational models inspired by the structure and function of the human brain. They consist ~~on~~ of interconnected nodes called neurons organised into layers.

2) Each neuron receives input from other neurons, processes the input and produces an output signal ~~and~~, which is then passed on to other neurons in the network.

3) perceptron →
It is based on the concept of single-layer feedforward neural network. It consists of a single layer of neurons, each of which receives input from a set of input features, performs a weighted sum of inputs, and applies an activation function to produce an output.



5) In the diagram, X's represent the input features, W's represent the weights and the bias is an addl. wt which is always set to 1.

The neuron computes the weighted sum of the inputs and bias and then applies an activation function to the sum to produce the output y

1) The perceptron is trained using a supervised learning algorithm called the perceptron ~~learning~~ learning rule.

During training, the perceptron is presented with a set of input/output pairs, and the weights are adjusted to minimize the difference between predicted and true output.

Weights are updated using the perceptron learning rule, ~~of~~ which adjusts the weights in proportion to the error between the predicted and true output.

6) It acts as the basis for more complex and powerful ANNs, such as ~~as~~ MLPs, CNNs and Recurrent Neural Networks (RNNs)

LOGISTIC REGRESSION :

1) It is a type of statistical model used to predict the probability of a binary outcome based on one or more predictor variables.

2) For eg. Lets say we want to predict whether a customer will buy a product based on their age, gender and income.
We can use log regression to build a model that predicts the probability of a customer buying the product based on these variables

3) In log regression, the log function is used to map the predicted values to probabilities, ranges from 0 to 1.

$$p = \frac{1}{1 + e^{-z}}$$

$$z = b_0 + b_1 x_1 + b_2 x_2 + \dots + b_n x_n$$

4) To fit a log regression model, we use a dataset with known outcomes to estimate the coefficients of the predictor variables that maximise the likelihood of the observed outcomes.
(maximum likelihood estimation)

5) Once we have estimated the coefficients, we can use the log function to predict the probability of a new customer buying the product based on their age, gender and income.

OVERFITTING :-

1) Overfitting is a common problem in ML. It occurs when a model is too complex and is fitted too closely to the training data. Hence, it may not generalize well to new data.

2) This is because, the model has learned to fit the noise of the training data, which may not be present in the test data

3) To detect overfitting, Splitting the data into training and validation sets is a technique. If model performance on training set increases while it deteriorates on the validation set, then it may be a sign of overfitting

REGULARIZATION :-

1) It is a technique used in machine learning to prevent overfitting of model.
2) Regularization adds a penalty term to the ~~loss~~ loss function during training, which prevents the model to learn over-complex relationships between ~~training~~ input features and output. and helps in generalization.
3) Types → L1 (Lasso regularization)
   → L2 (Ridge regularization)
4) L1 regularization →
   Adds the absolute values of the model's parameters to the loss function. This encourages the model to reduce the weights of less important features to zero, effectively performing feature selections.
5) L2 regularization →
   Adds the squares of the parameter values to the loss function. This encourages the model to reduce the magnitude of all parameters but not necessarily to zero.
6) It is usually used in combination with other techniques such as cross-validation to tune the hyperparameters of the model to improve generalization

Reinforcement Machine Learning :-

1) It is a type of machine learning algorithm that focuses on training agents to make decisions in an environment to maximize reward signa

2) The agent interacts with the environment over time and at each step, it receives a state observation from the environment and takes an action.

3) The environment then responds with a reward signal that indicates how well the agent performed and the process repeats.

~~4) It is applied to a wide range of problems~~

5) The goal of reinforcement learning ~~can~~ is to learn a policy, which is a mapping from states to actions that maximises cumulative output over time. The policy is learned through trial and error

6) ~~→~~ Categories → Model-based and Model-free

It is computationally expensive and requires careful tuning of hyperparameters.

Applications of Machine Learning :-

1. Fraud detection → fraudulent activity, unusual transaction patterns, multiple account access, changes in acc behavior
2. Predictive analytics → Analyze market data and predict future trends in stock prices, interest rates and forex rates
3. Algorithmic Trading → Develop trading algorithms, optimize portfolio performance, develop and backtest high-frequency trading strategies

Applications in FinTech :-

1. Fraud detection/prevention
2. Risk management → analyze credit risk by analyzing borrower characteristics
3. customer service → analyze customer data, provide personalized advice / products
4. Trading/ portfolio management
5. Credit Scoring → loan default, etc