Unit 3 Machine Learning

Supervised learning:-

* Supervised learning is a type of ML in which machines are trained using well-labeled training data & on the basis of that data, machines predict the output.
* Labelled data means some input data is already tagged with correct output.
* Training data → supervisor → teaches to predict o/p.

Types of Supervised ML:-

Supervised learning → Classification
Supervised learning → Regression

① Classification:-

* Classification is a type of supervised learning where the goal is to categorize or classify input data in to predefined classes or categories based on their features/past observations.

Ex:-

* It is a form of pattern recognition where the algorithm learns to distinguish between different classes or categories by analyzing features of input data.

Ex:-

① Classifying emails as either spam/not spam.
→ 0 or 1 → o/p is discrete.

② Classifying images of fruits-
→ 0/1/2 → o/p is discrete

③ Sentiment analysis →
→ positive/negative/neutral

④ Medical diagnosis →
→ disease present/no disease.

⑤ Customer retention → churn prediction.

Triveni TURBINES

Machine Learning

Unit 3: Supervised learning:-

* Supervised learning is a type of ML in which machines are trained using well-labeled training data & on the basis of that data, machines predict the output.
* Labelled data means some input data is already tagged with correct output.
* Training data → supervisor → teaches to predict o/p.

Types of Supervised ML:-

Supervised learning → Classification
Supervised learning → Regression

(1) Classification:-

* Classification is a type of supervised learning where the goal is to categorize or classify input data in to predefined classes or categories based on their features/past observations.

Triveni TURBINES

Ex:-

(1)

* It is a form of pattern recognition where the algorithm learns to distinguish bet? different classes or categories by analyzing features of input data.

Ex:-

(1) Classifying emails as either spam/not spam.
→ 0 or 1 → o/p is discrete.

(2) Classifying images of fruits-
→ 0/1/2 → o/p is discrete

(3) Sentiment analysis →
→ positive/negative/neutral

(4) Medical diagnosis →
→ disease present/no disease.

(5) Customer retention → churn prediction.

Triveni TURBINES

Key characteristics of a classification:-

1. Discrete o/p variable: Discrete → type of variable that has specific & separate values as opposed to continuous variables which can take any value within a range
   Ex:-
   - Gender classification → Male, Female
   - Loan approval → Approved, Rejected
   - Movie Genre classification → Action, Romance, Thriller, Comedy, Drama

2. Supervised learning:- labeled training data

3. Binary or multi-class classification:- (Two classes) or (more than 2 classes)
   Ex:- Binary classification:-
   Spam | Not spam

   Multi-class classification:- News categories
   Politics | Sports | Technology & entertainment

4. Decision boundaries:- Deciding boundaries to differentiate bet^n classes based on i/p features

5. Performance evaluation metrics:- Accuracy, precision, recall, F1 score.

Importance & benefits of classification in ML

1. Decision making:- Patterns, trends, relationship
2. Pattern recognition
3. Predictive modeling
4. Risk assessment:- Fraud detection
5. Personalization & recommendation systems
6. Image & speech recognition
7. Healthcare & biomedical research
8. Customer segmentation:- Demographics, behavior or preferences. It leads to tailoring marketing campaigns & increased retention rates.
9. Quality control & anomaly detection:- Defective/ non defective
10. Automated decision-making

Triveni TURBINES

Classification algorithms in ML

1. Logistic regression:-
   - Desc:- Linear classification algorithm. Probability that a given input belongs to a particular class.
   - Advantages:- Simple, interpretable,
   - Appl?s:- Spam detection, customer churn prediction.

2. Support vector machines (SVM):-
   - Desc:- Finds the optimal hyperplane to separate classes in feature space
   - Adv:- High dimensional spaces
   - Appl?s:- Text categorization, image recognition

3. Decision trees:-
   - Desc:- Non-linear classification algorithm that recursively split data based on feature values to make predictions.
   - Adv:- Easy to interpret, can handle both numerical & categorical data
   - Appl?s:- Customer segmentation, medical diagnosis

4. Random forest:-
   - Desc:- Consists of multiple decision trees. Aggregation of predictions of individual trees.
   - Adv:- Handles high-dimensional data, robust to overfitting.
   - Appl?:- Credit risk analysis, image classification

5. K-Nearest Neighbors (KNN):-
   - Desc:- Data points classified based on majority class among nearest neighbor
   - Appl?:- Pattern recognition, recommendation system
   - Adv:- Simple, easy to implement.

6. Neural networks:-
   - Desc:- Interconnected layers of nodes. Learns complex patterns through training with backpropagation.
   - Appl?:- Large datasets.
   - Adv:- Speech recognition, Image recognition.

Triveni TURBINES

Classification algorithms in ML

1. Logistic regression:-
   - Desc:- Linear classification algorithm. Probability that a given input belongs to a particular class.
   - Advantages:- Simple, interpretable,
   - Appl?s:- Spam detection, customer churn prediction.

2. Support vector machines (SVM):-
   - Desc:- Finds the optimal hyperplane to separate classes in feature space
   - Adv:- High dimensional spaces
   - Appl?s:- Text categorization, image recognition

3. Decision trees:-
   - Desc:- Non-linear classification algorithm that recursively split data based on feature values to make predictions
   - Adv:- Easy to interpret, can handle both numerical & categorical data.
   - Appl?s:- Customer segmentation, medical diagnosis

4. Random forest:-
   - Desc:- Consists of multiple decision trees Aggregation of predictions of individual trees.
   - Adv:- Handles high-dimensional data, robust to overfitting.
   - Appl?:- Credit risk analysis, image classification

5. K-Nearest Neighbors (KNN):-
   - Desc:- Data points classified based on majority class among k nearest neighbors
   - Appl?:- Pattern recognition, recommendation system
   - Adv:- Simple, easy to implement.

6. Neural networks:-
   - Desc:- Interconnected layers of nodes. Learns complex patterns through training with backpropagation.
   - Appl?:- Large datasets
   - Adv:- Speech recognition, Image recognition.

Triveni TURBINES

Performance evaluation metrics in classification:

Approach to measure performance is using confusion matrix.

Confusion matrix :- This table allows visualization of performance of an algorithm by displaying counts of true positive, true negative, false positive & false negative predictions.

Ex :- Confusion matrix for a binary classification problem

Actual / Predicted	Predicted Negative	Predicted Pos
Actual Negative	True Negative (TN)	False Positive
Actual Positive	False Negative (FN)	True Positive

Triveni TURBINES

TP :- Correctly classified as positive class

TN :- Correctly classified as negative class. (not belonging to positive class)

FP :- Incorrectly classified as positive class

FN :- " " " negative class

Using values in confusion matrix, we can calculate various metrics to evaluate the performance of classification algorithm.

(1) Accuracy :- Correct predictions made by the model out of the total number of predictions.

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$$

Accuracy can be misleading when class distribution is imbalanced

(2) Precision :- Proportion of true positives among the instances that the model predicted as positive

$$\text{Precision} = \frac{TP}{TP + FP}$$ → no. of instance predicted as +ve

Triveni TURBINES

(3) Recall: Proportion of true positives among the instances that are actually positive

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

[Useful when cost of false negatives is false]

(4) F1 score: When the class distribution is imbalanced,

$$\text{F1 score} = \frac{2 * (\text{precision} * \text{recall})}{\text{precision} + \text{recall}}$$

F1 ↑ model has both good precision & recall.

It is useful when both FP & FN are equally important

Triveni TURBINES

(1) Accuracy $= \frac{TP+TN}{TP+TN+FP+FN}$

$= \frac{20+65}{20+65+5+10} = 85\%$

(2) Precision $= \frac{TP}{TP+FP}$

$= \frac{20}{25} = 0.8 = 80\%$

Out of all instances predicted as +ve, 80% of them are actually +ve

(3) Recall (Sensitivity)

$$\text{Recall} = \frac{TP}{TP+FN} = \frac{20}{30} = 67\%$$

Out of all actual positive instances, the model is able to capture 67% of them. This means model is missing some of +ve instances

(4) F1 score

$$\text{F1 score} = \frac{2 * (\text{Precision} * \text{recall})}{\text{Precision} + \text{recall}}$$

$= \frac{1.072}{1.47} = 73\%$

Triveni TURBINES

Good balance bet^ precision & recall
This means model is making accurate true predictions while capturing majority of actual true instances
F1 score is lower than accuracy. →
Class distribution may be imbalanced

Regression:-

↳ It is a type of supervised learning technique in ML where the goal is to predict continuous or quantitative outputs based on i/p features. (real numbers)

Eg:

* If there is continuity bet^ possible outcomes, then the problem is regression problem.
* Regression is a process of finding correlations bet^ dependent & independent variables. It helps in predicting continuous variables.
* The objective of regression is to establish a relationship bet^ input variables & continuous target variable & allowing model to make predictions on new or unseen data points.

Triveni TURBINES

Ex:-

① Real estate valuation:-
Task :- Predict market value of properties
Features:- Area, no. of bedrooms, age of property, amenities etc.
Target variable :- Estimated price of property
Purpose:- Helps buyers & sellers get a fair idea of property prices & assists investors & real estate companies in making informed decisions

② Stock market prediction:-
Task:- Predict the stock price or return value

Triveni TURBINES

Key characteristics of a regression

(1) Continuous o/p variables: Continuous refers to o/p variable that can take any value within a range. These are quantifiable & can be subdivided into finer increments which are not restricted to separate categories

Ex - (1) House price prediction
(2) Stock price forecasting

House price prediction :- The o/p is real numbers that can vary across a wide range depending on i/p features

Stock price forecasting :- It is a numerical figure that represents stock price at a future time. They are continuous value that can fluctuate every minute

(2) Supervised learning :- Relies on labeled training data

(3) Linear Vs Non-linear :- Depending on nature of relationship bet" variables, it can be linear (simple linear regression) or non-linear (polynomial regression, logistic regression for binary outcomes in a continuous probability scale)

(4) Decision boundaries :- Line / curve that best fits the data points in the feature space

(5) Performance evaluation Metrics :-
(1) Mean Squared error (MSE)
(2) Root mean squared error (RMSE)
(3) Mean absolute error (MAE)
(4) R-squared (Coefficient of determination)

Triveni TURBINES

Types of regression

(1) Simple linear regression: It assumes linear relationship b/t independent variable & target variable
It predicts a response variable using a single feature
Ex: Predicting student's exam score based on no. of hours they studied.

(2) Multiple linear regression: It uses multiple features to predict a response
Ex: Predicting house prices based on features like square footage, no. of bedrooms, location & age of house
- How multiple features are considered for the prediction

(3) Polynomial regression: It models the relationship b/t independent variable & dependent variable as n^{th} degree polynomial, allowing for more complex curve fitting

Ex: Predicting the growth of a plant based on time spent in sunlight. This model will capture non linear relationships such as accelerated growth with increased sunlight exposure

(4) Logistic regression -

Regression algorithms in ML:

(1) Linear regression (2) Polynomial regression
(3) Ridge (4) Lasso regression
(5) Random forest regression
(6) Support vector regression
(7) Decision tree regression
(8) Gradient boosting regression

Triveni TURBINES

Importance & benefits of regression in ML

1. Predictive modeling: Prediction & forecasting future trends.

2. Interpretability: Linear regression provide coefficients that indicate the impact of each feature on the target variable. This helps in understanding driving factors behind predictions

3. Feature selection: Identifies most important features that influence target variable. Lasso regression technique selects features by shrinking coefficients to 0 leading to more relevant model

4. Model Evaluation: Evaluation Metrics.

5. Handling non-linear relationships: Polynomial regression, decision tree regression & support vector regression can capture non-linear relationships bet^n variables

Triveni TURBINES

6. Risk assessment: Stock prices, Credit risk & insurance claims

7. Optimization & decision making: Determining optimal pricing strategy, resource allocation or process improvement based on predictive insights

8. Scalability & efficiency: Regression algorithms can handle large datasets efficiently making them suitable for high-dimensional data

9. Generalization: Generalize well on unseen data

10. Versatility: Regression techniques can be applied to various domains such as healthcare, marketing, finance & engineering

Triveni TURBINES

Performance evaluation metrics

① Mean Squared error (MSE):
Average of squared differences bet° predicted values & actual values.

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

y_i = actual value
$\hat{y}_i$ = Predicted value

* Model accuracy
* Sensitive to outliers
* Use it when large errors are bad & you want to emphasize them

② Root mean squared error (RMSE):
* Square root of MSE
* Measure of standard deviation of residuals
* $RMSE = \sqrt{\frac{\sum (y_i - \hat{y}_i)^2}{n}}$
* Interpretable in the same units as target variable
* Sensitive to outliers

Triveni TURBINES

③ Mean absolute error (MAE)
Calculates the average of absolute differences bet° predicted values & actual values

$$MAE = \frac{\sum_i |y_i - \hat{y}_i|}{n}$$

Less sensitive to outliers as compared to MSE & RMSE, Use it when you want more balanced view of average error

④ R-squared (R^2):- Measures the proportion of variance in the dependent variable that is predictable from the independent variables

$$R^2 = 1 - \left[\frac{\sum (y_i - \hat{y}_i)^2}{\sum (y_i - \bar{y})^2}\right]$$

Range = 0 (worst), 1 (best)

Higher R^2 → better fit of the model to the data.

(5) Adjusted R Squared :-

$$\text{Adjusted } R^2 = \frac{1-(1-R^2)*(n-1)}{n-p-1}$$

Preventing overfitting by considering number of predictors

(6) Mean Squared Logarithmic Error (MSLE):

Useful when target variable has exponential growth pattern

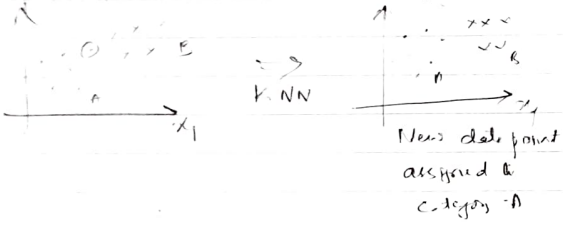
$$MSLE = \frac{\sum (\log(1+y_i) - \log(1+\hat{y}_i))^2}{n}$$

Triveni TURBINES

K - Nearest Neighbors algorithm

* Suppose there are 2 categories, category A & category B & we have a new data point x_1. Question is in which of these categories will it lie?

* To solve this we need K NN algorithm



* Based on the closest resemblance or similarity to features of known cat and dog, algorithm will classify new image as either cat or dog category

* It is used for both classification & regression tasks.

Triveni TURBINES

Def: K-NN algorithm is a fundamental ML technique that operates on the principle of similarity.

A new data point's class is determined by the majority class of its nearest neighbors in the training set.

Characteristics of K-Nearest Neighbors (K-NN) algorithm

1. Non-parametric nature: It relies on the similarity of data points for classification or regression.

2. Lazy learning approach: It postpones the learning process until the classification phase. It stores the training data & performs classification only when a prediction is requested.

3. Storage of training data:-

④ Nearest neighbor classification:-
The majority class among these neighbors is assigned to new data point for classification.

⑤ Regression & Classification:-
Average of target values of its nearest neighbors.

⑥ Similarity-based prediction:-
K-NN operates on the assumption that similar data points belong to the same class or have similar target values.

⑦ Versatility & Ease of implementation:-
Data distribution is not explicitely defined.

How K-Nearest Neighbors (K-NN) works?

① Store training data:- Without performing any computation on the data.

② Calculate distance:-
When a new data point is presented for prediction, the algorithm calculates the distance bet? new data point & all the data points in the training set.
Typically Euclidean distance is used.

③ Select nearest neighbors:-
After calculating distance, the algorithm identifies K-nearest data points to the new data point based on the calculated distances.
K is a hyperparameter that needs to be predefined by the user. K → tells no. of neighbors considered for classification or regression.

④ For Classification:-

Once the k-nearest neighbors are identified, the algorithm assigns the majority class label among these neighbors to the new data point.

⑤ For Regression:-

The algorithm calculates the average of the target values of K-Nearest neighbors. This average value is the predicted value.

K-NN algorithm:-

Step 1:- Choose the number of neighbors (K) before running the algorithm.

Step 2:- Calculate distance.

Euclidean distance measures the similarity or proximity bet? data points in the feature space

Step 3:- Sort & Select Nearest neighbors:-

After calculating the distances, sort the distances in ascending order & selects the K-data points with the smallest distances to new data point. These K data points are the nearest neighbors to new data point in the feature space.

Step 4:- For classification:- The class with the highest frequency among K-neighbors is chosen as predicted class for new data point.

Step 5:- For regression:- Compute the average of target values of K-nearest neighbors.

How to select value of K in K-NN algorithm?

Start with small values of K (eg, K=3 or K=5) & gradually increase the value while monitoring the model's performance. This iterative approach will help in finding an optimal K value that balances bias & variance in the model.

How to calculate Euclidean distance?

For 2 points (2D):- Distance bet^n 2 points in 2 dimensional space (2D)

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

For 3 points (3D):-

$$d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2 + (z_2 - z_1)^2}$$

For 4 points (4D):-

$$d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2 + (z_2 - z_1)^2 + (w_2 - w_1)^2}$$

$x_1, y_1, x_2, y_2, z_1, z_2, w_1, w_2$ are co-ordinates of data points in spaces.

Example 1: KNN algorithm for classification task

Dataset of fruits → Classifying fruits based on 2 features - Sweetness & acidic.

Now we need to classify new fruit based on its sweetness & acidic values.

Training data

~~Fruit~~	Sweetness	Acidity	Type
Fruit 1	8	3	Apple
2	6	2	Apple
3	3	7	Lemon
4	2	8	Lemon

New data point:- Sweetness 5, Acidic 4.

Find the type of fruit using KNN algorithm.

Triveni TURBINES

Sol?: Let K = 3.

Calculate euclidean distance bet? new data point & all data points in the training set.

Dist from (5, 4) to fruit 1 (8, 3) =

$\text{Sqrt}\left((8-5)^2+(3-4)^2\right) = 3.16$

$\text{Sqrt}\left((6-5)^2+(2-4)^2\right) = 2.24$

$\text{sqrt}\left((8-5)^2+(7-4)^2\right) = 3.61$

$\text{sqrt}\left((2-5)^2+(8-4)^2\right) = 5.$

Sort in ascending order.

fruit 2
fruit 1
fruit 3
fruit 4.

Select fruit 2, fruit 1 & fruit 3 (K = 3)

Select Majority class :- Apple

Triveni TURBINES

Example 2

A	2	3	Class 1
B	3	5	"
C	3	2	Class 2
D	6	7	"

E: (4, 4).

E, A = 2.24
E, B = 1.41
E, C = 1.41
E, D = 3.16.

Class 1

Triveni TURBINES

Example 3: KNN algorithm for regression task.

Use KNN algorithm to predict price of new fruit based on its sweetness & acidic value.

Training set

	Sweetness	Acidic	Price
Fruit 1	8	3	100
" 2	6	2	80
3	3	7	50
4	2	8	40

New data point : Sweetness: 5
Acidic : 4.

Let K=3,

Distance from (5,4) to (8,2) = 3.16.
" " " " (6,2) = 2.24
" " " " (3,7) = 3.61
" " " " (2,8) = 5

Triveni TURBINES

Fruit 2
" 1
" 3
" 5
} Selected

Predicted price = (Fruit2 price + Fruit1 price + Fruit3 price)/3

$= \frac{80+100+50}{3} = 76.67.$

Example 4:- Classifying students based on 3 features Math, CS & English scores.
Based on new student's Math, CS & English scores, we will classify him/her either fail/Pass

	Math	CS	English	o/p
I1	4	3	2	Fail
I2	6	7	5	Pass
I3	7	8	6	Pass
I4	5	5	4	fail
I5	8	8	7	Pass

New data pt:- Maths - 6
CS - 8
English - 6

Triveni TURBINES

$k = 3$

DIS: $(6, 8, 6)$ to $(4, 3, 2) = 6.71$

" " $(6, 7, 5) = 1.41$

" " $(7, 8, 6) = 1$

" " $(5, 5, 4) = 3.74$

" " $(8, 8, 7) = 2.24$

1, 1.41, 2.24

I1, I2, I5

Majority = Pass

Applications of KNN algorithm:

① Healthcare :- Disease diagnosis & patient outcome prediction.
Analysis of medical history & symptoms of patients, KNN helps in identifying similar cases & recommending appropriate treatments or interventions.

Triveni TURBINES

② Finance :- Credit scoring, fraud detection & stocks market analysis. Compares financial behavior of customers or detects unusual patterns in transactions.

③ Retail :- Customer segmentation, personalized recommendations & market basket analysis.
Identifying similar customer profiles or recommending products based on past purchases.

④ Social media :- Friend recommendations, content filtering & sentiment analysis.
Identifies user interactions & preferences, KNN suggests connections, filters news feeds & categorizes user sentiments to enhance user engagement.

⑤ Environmental Science :- Species classification, pollution monitoring & climate modeling.
Predict species distribution, detect pollution hotspots & model climate changes.

Triveni TURBINES

Advantages

1. Simple to implement
2. Easy to understand & gives reasonable performance without a lot of tuning
3. It is robust to noisy training data
4. It can be more effective if training data is large.
5. Building nearest neighbors model is typically fast, although prediction speed may decrease with very large training sets.

Disadvantages:

1. Need to determine value of K can sometimes be challenging
2. Computation cost is high
3. It doesn't perform well on datasets with many features.
4. Difficulty in proper classification when dealing with high-dimensional data.

Triveni TURBINES

2. **Linear Models** :- These are fundamental class of algorithms in supervised machine learning that make predictions by computing linear combinations of i/p features.

In linear model, the relationship bet? i/p features & target variable is represented as linear f?

$$Y = C_0 + C_1X_1 + \ldots + C_nX_n$$

↳

1. Y (Dependent variable) :- It is the value to be predicted. (Target variable or response variable)

2. X (Independent variables) :- These are used to predict Y. (Features)

Ex:

$X_1 \rightarrow$ size of house
$X_2 \rightarrow$ no. of bedrooms
$X_3 \rightarrow$ age of house
$Y \rightarrow$ house price

③ C (coefficients) :- These coefficients / parameters quantify the relationship bet? each independent variable & dependent variable.

C_0 → Special coefficient (intercept) represents the expected value of Y when all X = 0.

$C_1, C_2, C_3 \cdots C_n$ → slopes for the respective X variables. They represent how much Y is expected to change with a one unit change in the corresponding X variable, holding all other variables constant.

Classification & regression tasks with linear models:-

① Regression with linear models:- Here linear models predict a continuous target variable based on i/p features by fitting a linear relationship bet? features & target.

Ex 1 :- In house price prediction :- linear regression model learns coefficients for each feature & an intercept term.

Ex 2 :- Predicting student scores based on study hours. Algorithm :- linear regression.

② Classification with linear models :- These models separate classes by defining a linear decision boundary in the feature space to classify data points into different categories.

Ex 1 :- Spam email detection system (keywords & sender information) → binary classification.

Transaction is fraudulent / not → based on transaction features like amount, location & time.

Algorithm :- Logistic regression

Characteristics of linear models:-

① Linear relationships :- Linear models assume a linear relationship bet? input features & target variables. The predicted o/p is a linear combination of i/p features.

$Y = C_0 + C_1 * Area + C_2 * Bedrooms + C_3 * Location$

Triveni TURBINES

Ex:-

② Intrepretability :- Linear models provide interpretable coefficients that indicate the impact of each feature on target variable.
Ex:- In logistic regression for spam email detection, a positive coefficient for the keyword feature indicates that the presence of that keyword increases the likelihood of an email being classified as spam.

③ Scalability :- Training & making predictions with linear models are generally faster as compared to other models due to their capability to handle large datasets with many features.
Ex:- Sentiment analysis task with a large text dataset using SVM.

④ Regularization :- Techniques like Lasso (L1) & Ridge (L2) regression help in improving model's generalization.

Triveni TURBINES

⑤ Binary & multi-class classification :- Linear models can be used for both binary & multi-class classification tasks. For binary classification, logistic regression is used while strategies like One-Vs-Rest or One-Vs-One can extend linear models for multi-class classification.
Ex:- In a medical diagnosis system, linear models can classify patients into multiple disease categories based on various medical test results.

① Linear regression :

How linear regression works?

① Model representation :-

$$Y = C_0 + C_1 X_1 + \dots + C_n X_n$$

② Goal is to find values of co-efficients $C_0, C_1, C_2, C_3 \dots$ that can minimize the difference bet^n predicted values & actual target values.

③ Training :- The algorithm learns the optimal values of coefficients by minimizing MSE.

④ Prediction :- Once trained, the model can make predictions on new data by putting i/p features into learned eq^n.

Triveni TURBINES

Ex: Car price prediction using linear regression

Linear regression model → car price prediction based on its age & Mileage.

Car price = $C_0 + C_1 \times$ Mileage $+ C_2 \times$ Age
(Y) (Lakhs of INR) (Miles) (years)

C_0 → base price of car.
C_1 → how much car price decreases for every additional thousand miles driven.
C_2 → how much car price decreases for every additional years of age.

Given data sample.

Mileage (×1000 km)	Age (years)	Price (lakhs)
80	5	5
50	2	7
90	4	4.5
30	1	8
70	3	6

Triveni TURBINES

Model training & Coefficients :-

By training we got $C_0 \Rightarrow$ 9.75 lakhs. → estimated base price of car with 0 Mileage & 0 age

$C_1 = -0.05$. → for every additional 1000 km, price decreases by 0.05 lakhs (50,000)

$C_2 = -0.15$ → for every additional year, price decreases by 0.15 lakhs (15,000)

New data: car with 45,000 km mileage & 3 years old.

Predicted car price = $C_0 + C_1 \times$ Mileage $+ C_2 \times$ Age
$= 9.75 + (-0.05 \times 45) + (-0.15 \times 3)$
$= 7.05$ lakhs.

Triveni TURBINES

Ex. Home price prediction using linear regression. (Based on its size in square feet) & age.

House Price = $C_0 + C_1 * \text{Size} + C_2 * \text{Age}$
(in lakhs) (sq. feet) (years)

Size (sq. ft)	Age (years)	Price (in lakhs)
1500	5	200
1200	2	180
1800	4	220
1000	1	160
1400	3	190

Calculate C_0, C_1 & C_2.

Step 1: Calculate mean values

Mean Size = $\frac{1500+1200+1800+1000+1400}{5}$ = 1380 sq.ft

Mean age = $\frac{5+2+4+1+3}{5}$ = 3 years

Mean price = $\frac{200+180+220+160+190}{5}$ = 190 lakhs.

Triveni TURBINES

Step 2: Calculate variance & covariance

$$\text{Covariance}(\text{Size}, \text{Price}) = \frac{\sum((\text{Size} - \text{Mean Size}) \times (\text{Price} - \text{Mean Price}))}{(n-1)}$$

$$= \frac{(1500-1380)(200-190) + (1200-1380)(180-190) + (1800-1380)(220-190) + (1000-1380)(160-190) + (1400-1380)(190-190)}{4}$$

$$= 6740.$$

$$\text{Covariance}(\text{Age}, \text{Price}) = \frac{\sum((\text{Age} - \text{Mean Age}) * (\text{Price} - \text{Mean Price}))}{n-1}$$

$$= 30.$$

$$\text{Variance}(\text{Size}) = \frac{\sum((\text{Size} - \text{Mean Size})^2)}{n-1}$$

$$= 92000$$

$$\text{Variance}(\text{Age}) = \frac{\sum((\text{Age} - \text{Mean Age})^2)}{n-1}$$

$$= 2.5$$

0.073 × 100000

Triveni TURBINES

Step 3: Calculate the Coefficients.

$C_1 = \frac{\text{Covariance(Size, Price)}}{\text{Variance(Size)}} = 0.073$ (For every one square feet increase in size of house, house price is expected to increase by 7300)

$C_2 = \frac{\text{Covariance(Age, Price)}}{\text{Variance(Age)}} = 12$ (For every one year increase in age of house, predicted price increases by 12 lakhs.)

$C_0 = \text{Mean Price} - C_1 \times \text{Mean Size} - C_2 \times \text{Mean age}$

$= 190 - 0.073 \times 1380 - 12 \times 3 = 53.26$ lakhs.

Step 4: New data pt :- 1300 Sq. ft, 2 years old

$\text{House Price} = C_0 + C_1 \times \text{Size} + C_2 \times \text{Age}$

$= 53.26 + 0.073 \times 1300 + 12 \times 2$

$= 172.16$ lakhs.

Triveni TURBINES

(2) Logistic regression:- It is a classification algorithm that predicts the probability of an observation belonging to a certain class.

* It uses logistic function to map the o/p to a probability value bet^n 0 & 1.

Ex:- (1) Binary classification using logistic regression Predicting student pass or fail based on no. of hours studied

(1) Data

Independent variable:- No. of hours studied

Dependent :- Pass(1) or Fail(0)

(2) Logistic regression model:- o/p will a probability value bet^n 0 & 1.

(3) Prediction:- Given a new student who has studied for 5 hours, the model predicts the probability of passing exam as 0.8.

(4) Classification:- If the predicted probability is greater than a chosen threshold (eg. 0.5) student is classified as Pass, otherwise fail.

Triveni TURBINES

Ex: ② Multi-class classification using multinomial logistic regression.

Classify images of fruits in to 3 categories. apples, oranges, & bananas based on their size & color.

Data

↳ Independent variable (X):- Size & color features of fruits.

Dependent variable (Y):- Apples, oranges, bananas.

Multinomial logistic regression model:-

Model's o/p will include the predicted probabilities for each class.

Prediction:-

Given a new fruit with specific size & color features, model predicts the probabilities of it being an apple, orange & banana.

Classification:- Fruit is classified into the category with the highest predicted probability.

Triveni TURBINES

Applications of linear models

① Predictive Modeling:- House price, sales forecast prediction, spam/not spam.

② Marketing & business:- Market basket analysis, customer segmentation

③ Finance:- Predicting credit risk, loan default probabilities, insurance claim likelihood etc, Stock market analysis.

④ Healthcare:- disease prediction, drug response prediction.

⑤ Recommendation Systems

⑥ Natural language processing:- Sentiment analysis, spam detection, text categorization task.

Triveni TURBINES

Advantages of Linear models

1. Easy to interpret & understand. Coefficients provide insights into relationship bet? input features & target variables.
2. Large dataset computations is efficient in linear models.
3. Linear models help in identifying most important features influencing target variables.
4. Linear models serve as good baseline model for more complex algorithms, helping to compare performance.

Disadvantages:

1. Linear models assume a linear relationship bet? features & target variable which may not hold true for ~~large~~ complex datasets.
2. Linear models may not capture intricate relationships in the data compared to non-linear models like decision trees or neural nfws.

Triveni TURBINES

3. Outliers can significantly impact the coefficients in linear models.
4. Handling categorical variables in linear models may require encoding techniques like one-hot encoding which can increase dimensionality.

Naive Bayes Classifiers:-

* Popular algorithm in natural language processing tasks like spam filtering & document categorization.

*. It is a probabilistic method for categorizing data points based on Bayes' theorem which establishes connection bet? present data & existing assumptions or beliefs.

Triveni TURBINES

Bayes' Theorem:- It describes the probability of an event, based on the prior knowledge of conditions that might be related to the event.

* In the context of classification, Baye's theorem is used to calculate the probability of a class label given the observed features.

$$P(A|B) = \frac{P(B|A) \times P(A)}{P(B)}$$

($P(A|B)$ → posterior probability; $P(B|A)$ → likelihood; $P(A)$ → prior; $P(B)$ → evidence)

where

$P(A|B)$:- The probability of hypothesis A given the evidence B. (Posterior Probability)

$P(B|A)$:- The probability of evidence B given the hypothesis A. (Likelihood probability)

$P(A)$:- Prior probability of hypothesis A.

$P(B)$:- Prior probability of evidence B.

Triveni TURBINES

Ex:- Medical diagnosis.

Scenario:-

Hypothesis (A):- A patient has a particular disease.

Evidence (B):- The results of a diagnostic test for the disease.

Probabilities:-

Prior Probability ($P(A)$):- Prior probability that the patient has the disease = 0.01 (1% of population has the disease)

Likelihood ($P(B|A)$):- 0.95 (95% chance of true test if patient has disease)

Evidence Probability ($P(B)$):- 0.02 (2% false tve rate)

Calculation:-

$$P(A|B) = \frac{P(B|A) \times P(A)}{P(B)}$$

$$= \frac{0.95 \times 0.01}{0.02} = 0.475.$$

There is 47.5% chance that the patient actually has the disease given that they

have tested true on diagnostic test.

Ex 2:- Email Spam probability

Problem:- Consider an email system where
* 30% of emails are spam.
* If an email is spam, 40% chance it contains word 'free'
* If an email is not spam, 10% chance it contains word 'free'.

Calculate the probability that an email is spam given that it contains the word "free".

Scenario:-
Hypothesis (A):- An email is spam.
Evidence (B):- Email contains word "free"

Probabilities:-
P(A) = P(Spam):- 30% → Prior probability
P(B|A) = P(Free|Spam):- 40% → Likelihood
P(B) = P(Free):- Probability that an email contains word free

Calculation to determine P(B):-

$$P(Free) = P(Free|Spam) \times P(Spam) + P(Free|Not\ spam) \times P(Not\ Spam)$$
$$= 0.40 \times 0.30 + 0.10 \times 0.70$$
$$= 0.12 + 0.07$$
$$= 0.19$$

Calculate P(A|B) using Baye's theorem:-

$$P(Spam|Free) = (P(Free|Spam) \times P(Spam))/P(Free)$$
$$= (0.40 \times 0.30)/0.19$$
$$= 0.6315.$$

The probability that an email is spam given that it contains word free is 63.15%.

Naive Bayes Classifier:- Assumes independence between features.

* Effect of one feature on the class is independent of the presence of other features

Ex:- Text classification task → discount & offer words presence → spam.

The independence assumption implies that the occurrence of the word 'discount' in an email doesn't affect the occurrence of the word "offer" in the same email when determining if email is spam or not.

Naive & Bayes:-

Naive:- Occurrence of a certain feature is independent of occurrence of other features.

Bayes:- Depends on Bayes theorem.

Ex:- Naive Bayes Classifier:-

① Fruit classification

Features:- Color (red, Yellow), shape (Round, oval)

Training data:- Red, round → apple
yellow, oval → Orange

Prediction:- Fruit is red & round, classify as apple or orange using Naive Bayes.

② Email spam detection:-

Features:- Words in the email.

Training data:- Spam email:- "Get rich quick money", "huge discount offer", "free iphone"
Non-spam email:- Meeting scheduled for tomorrow at 10 AM.

Prediction:- Classify as spam or non-spam using Naive Bayes.

How Naive Bayes classifier works?

① Bayes' Theorem:-

$$P(A|B) = \frac{(P(B|A) \times P(A))}{P(B)}$$

$P(A|B)$:- Probability of class A given data B.
$P(B|A)$:- " " data B given class A
$P(A)$:- Prior probability of class A
$P(B)$:- " " " class B

Triveni TURBINES

(2) Naive Bayes Assumption:-

Naive Bayes simplifies the computation of $P(B|A)$ by assuming that all features in B (such as words in an email) are independent of each other given class A. This simplifies calculation.

(3) Classification process:-

Given a set of features $X = \{x_1, x_2, \dots x_n\}$ & a set of class labels $C = \{c_1, c_2, \dots c_k\}$, the Naive Bayes classifier predicts the most probable class label for i/p features.

* The classifier calculates the posterior probability for each class label & selects the class with highest probability.

(4) Model training – Calculating probabilities.

Calculate prior probabilities :- Probability of each class in training dataset like $P(A)$, $P(B)$ etc.

Triveni TURBINES

Calculate likelihoods $P(B_i|A)$:- Calculating the probability of each feature B_i given each class A.

(5) Calculating likelihood product:-

* Given a set of features $X = \{x_1, x_2, x_3 \dots x_n\}$, the likelihood of features given a class C_k is calculated by multiplying probabilities of each independent feature.

$$P(X|C_k) = P(x_1|C_k) * P(x_2|C_k) \times \dots \times P(x_n|C_k)$$

$P(x_i|C_k)$ → Probability of feature x_i given class C_k.

(6) Calculating probability of features $P(X)$

Generic formula for the total probability of a feature set X in the context of Naive Bayes classification is given by:

$$P(X) = P(X|c_1) \times P(c_1) + P(X|c_2) \times P(c_2) + \dots + P(X|c_k) \times P(c_k)$$

Where,

$P(X|C_i)$ → probability of observing feature set X given class C_i.

$P(C_i)$ → prior probability of class C_i.

k → total no. of classes.

(7) Calculating posterior probabilities for classification

$$P(C_k|X) = (P(X|C_k) \times P(C_k)) / P(X)$$

(8) Decision rule:

Selects class label C_k that maximizes posterior probability $P(C_k|X)$

Ex: Spam email detection:

Presence of word: "free" & "money"

* Spam emails (Total = 100)
  - free : 40 occurrences
  - money : 30 "

* Non-spam emails (Total = 100)
  - free = 10
  - money = 20

* Prior Probabilities
  - 50% of emails are spam
  - " " " " not spam.

We receive a new email that contains both "free" & "money". Classify it as either spam/not spam using Naive Bayes classifier.

Sol?:- Given data:-

Class C: $\{C_1, C_2\}$ => {Spam, Not spam}

Features X: $\{x_1, x_2\}$ => {"free", "money"}

Total Spam emails (Spam): 100

Total Non-spam emails (Not spam): 100

Occurrences of "free" in spam emails: 40

" " "money" " " " : 30

" " "free" in Non-spam emails: 10

" " "money" in " " : 20

Prior probability of spam (P(Spam)): 0.5

" " : Not spam (P(Not Spam)): 0.5

Triveni TURBINES

Step 1: Calculate likelihoods.

Likelihood of "free" given spam (P(free|spam)):
$\frac{40}{100} = 0.4$.

Likelihood of "money" given spam (P(money|spam)):
$\frac{30}{100} = 0.3$.

Likelihood of "free" given Non-Spam (P(free|not spam))
$= \frac{10}{100} = 0.1$

Likelihood of "money" given Non-spam (P(money|not spam))
$= \frac{20}{100} = 0.2$

Step 2: Calculating likelihood Product:-

Posterior probability for each class (Spam or not spam) should be calculated.

Triveni TURBINES

The likelihood product $P(X|C_k)$ is calculated by assuming feature independence.

$$P(X|C_k) = P(x_1|C_k) \times P(x_2|C_k) \times \dots \times P(x_n|C_k)$$

$$P(X|\text{Spam}) = P(\text{free}|\text{spam}) \times P(\text{money}|\text{spam})$$
$$= 0.4 \times 0.3 = 0.12.$$

$$P(X|\text{not spam}) = P(\text{free}|\text{not spam}) \times P(\text{money}|\text{not spam})$$
$$= 0.1 \times 0.2 = 0.02.$$

Step 3: Calculating probability of features P(X)

Total probability of all features (P(X))

$$P(X) = P(X|\text{spam}) \times P(\text{spam}) + P(X|\text{not spam}) \times P(\text{not spam})$$
$$= (0.12 \times 0.5) + (0.02 \times 0.5)$$
$$= 0.06 + 0.01$$
$$= 0.07.$$

Triveni TURBINES

Step 4 :- Calculating posterior probabilities for classification. (Given set of features belongs to class C_k)

Bayes' theorem.

$$P(C_k|X) = \frac{(P(X|C_k) \times P(C_k))}{P(X)}$$

$$P(\text{spam}|X) = \frac{(P(X|\text{spam}) \times P(\text{spam}))}{P(X)}$$

$$= \frac{0.12 \times 0.5}{0.07}$$

$$= 0.8571$$

$$P(\text{not spam}|X) = \frac{P(X|\text{not spam}) \times P(\text{not spam}))}{P(X)}$$

$$= \frac{0.02 \times 0.5}{0.07}$$

$$= 0.142.$$

Triveni TURBINES

Step 5 : Decision rule :- Spam. (class with highest probability).

Triveni TURBINES

Decision trees:-

Decision tree-based algorithms use a tree-like model to make decisions based on input data.

It consists of series of nodes that represents decisions or tests on i/p data & branches that represent possible outcomes of those decisions/tests. Leaves of tree → final decision or Prediction.

*. Selecting best attribute to split data at each node, based on measure of info gain or impurity reduction.

*. ID3, C4.5 & CART → algorithms.

Decision tree algorithm

* Predictive modeling & classification tasks.

Triveni TURBINES

Decision tree algorithm:-

1. Begin tree with root node, S, which contains complete dataset.
2. The best attribute that provides best split for the dataset is found using Attribute selection measure (ASM) → information gain, gain ratio & gini impurity.
3. Divide the dataset S into subsets that contain possible values for the best attribute found in step 2. Each subset represents a branch of decision tree based on the values of selected attribute.
4. Generate a decision tree node (internal node) that contains best attribute.
5. Recursively make new decision trees using subsets of dataset created in step 3. Continue this process until a stage is reached where we cannot further classify nodes & then nodes are leaf nodes. Final outcome is represented by leaf nodes.
6. The process continues until the entire dataset is classified into their respective categories at leaf node.

* This algorithm is used to construct a decision tree that can be used for classification & prediction tasks.

Attribute Selection Measure (ASM)

* It is a criterion used in decision tree algorithms to select best attribute for splitting data at each node.

* ASM assigns a score to each attribute based on its ability to divide data into subsets that are more homogeneous in terms of target variable.
* ASM with highest score → splitting attribute

(i) Information gain:- To calculate information gain, we need to calculate entropy. It is a measure of impurity or randomness of dataset & it is used to measure effectiveness of attributes in dividing data in to more heterogeneous subsets with respect to target variable

$$Gain(A) = Entropy(D) - Entropy_A(D)$$

$$Entropy_A(D) = \sum_{j=1}^{v} \frac{|D_j|}{|D|} \times Entropy(D_j)$$

where

$D_1, D_2, \ldots D_k$ are subsets of D that correspond to each value of attribute A

$|D_1|, |D_2|, \ldots |D_v|$ are sizes of those subsets.

$Entropy_A(D)$ → Average entropy after ~~Entropy~~ partitioning dataset based on values of attribute A.

$$Entropy(D) = -\sum_{i=1}^{m} p_i \log_2 p_i$$

p_i → probability that an arbitrary tuple in D belongs to class C_i.

$$Entropy(D) = -p(Yes)\log_2 p(Yes) - p(No)\log_2 p(No)$$

↳ for binary classification

Lower entropy level → pure dataset

higher " " → impure "

Triveni TURBINES

Problem: Suppose we have a dataset of 10 examples, each with binary class label ("Yes" or "No"). 6 → Yes, 4 → No. The entropy of dataset is

$$\text{Entropy}(D) = -\frac{6}{10}\log_2\left(\frac{6}{10}\right) - \frac{4}{10}\log_2\frac{4}{10}$$

$$\text{Entropy}(D) = 0.971$$

↳ Impure/random

Information gain problem

Example	Outlook	Temperature	PlayTennis
1	Sunny	Hot	No
2	Sunny	Hot	No
3	Overcast	Hot	Yes
4	Rainy	Mild	Yes
5	Rainy	cool	Yes
6	Rainy	cool	No
7	Overcast	cool	Yes
8	Sunny	Mild	No
9	Sunny	cool	Yes
10	Rainy	Mild	Yes

Triveni TURBINES

Goal:
We need to determine which attribute ("Outlook" or "temperature") is best to split on based on information gain.

Step 1: Entropy of entire dataset based on Play Tennis label.

PlayTennis = Yes (6) PlayTennis = No (4)

$$\text{Entropy}(D) = -\frac{6}{10}\log_2\left(\frac{6}{10}\right) - \frac{4}{10}\log_2\left(\frac{4}{10}\right)$$

$$\text{Entropy}(D) = 0.9709.$$

Step 2: Calculate information gain for attribute "Outlook".

> Information Gain (Outlook) = Entropy (D) − Weighted Average entropy

$$\text{Weighted Average Entropy} = \frac{4}{10} \times \text{Entropy}(D_1) + \frac{3}{10} \times \text{Entropy}(D_2) + \frac{4}{10} \times \text{Entropy}(D_3)$$

Triveni TURBINES

$$Entropy(D_1) = Entropy(Sunny) = -\frac{1}{4}\log_2\left(\frac{1}{4}\right) - \frac{3}{4}\log_2\left(\frac{3}{4}\right)$$

$$= 0.8113$$

$$Entropy(D_2) = Entropy(Overcast) = 0.$$

$$Entropy(D_3) = Entropy(Rainy) = -\frac{3}{4}\log_2\left(\frac{3}{4}\right) - \frac{1}{4}\log_2\left(\frac{1}{4}\right)$$

$$= 0.8113$$

So,

Weighted average entropy $= \frac{4}{10} \times 0.8113 + \frac{2}{10} \times 0 + \frac{4}{10} \times 0.8113$

$$= 0.3245 + 0 + 0.3245$$

$$= 0.649$$

Information gain for outlook = Entropy(D) − Weighted Entropy

$$= 0.97095 - 0.649$$

$$= 0.32195$$

Triveni TURBINES

Step 3: Calculate information gain for attribute Temperature

$$Entropy_{Hot} = -\frac{1}{3}\log_2\left(\frac{1}{3}\right) - \frac{2}{3}\log_2\left(\frac{2}{3}\right)$$

$$= 0.9183$$

$$Entropy_{Mild} = 0.9183$$

$$Entropy_{Cool} = -\frac{3}{4}\log_2\left(\frac{3}{4}\right) - \frac{1}{4}\log_2\left(\frac{1}{4}\right)$$

$$= 0.8113$$

Weighted entropy = 0.8755

Gain (Temp^r) = 0.97095 − 0.8755 = 0.09545

Conclusion

Outlook has higher info gain, it is the better attribute to split on for root of decision tree

The decision tree would look like this −

If outlook = Sunny then:

If temp^r = Hot then PlayTennis = No.

= Mild = No

= Cool = Yes

If outlook = Overcast, PlayTennis = Yes

If outlook = Rainy, then . . .

(ii) Gain ratio:- It measures effectiveness of a particular attribute in classifying data.

$$\text{Gain ratio}(A) = \frac{\text{Gain}(A)}{\text{SplitInfo}_A(D)}$$

where

Gain(A) => Information gain of attribute A on dataset D.

$\text{SplitInfo}_A(D)$ → Split info of attribute A on dataset D.

$$\text{SplitInfo}_A(D) = -\sum_{j=1}^{v} \frac{|D_j|}{|D|} \times \log_2\left(\frac{|D_j|}{|D|}\right)$$

v → no. of discrete values in A attribute

$\frac{|D_j|}{|D|}$ → weight of j^{th} partition.

Split information measures the uncertainity caused by different splits on "Study hours" attribute

It considers no. of study hour ranges & how many students fall into each range

Gain ratio will help in evaluating whether split is worth it.

Gain ratio Calculation

Date :

① Dataset

Student	Study hours	Pass/Fail
Alice	2	Fail
Bob	1	Fail
Carol	3	Pass
Dave	2	Fail
Eve	4	Pass

② Entropy

$$\text{Entropy}(D) = -\frac{2}{5}\log_2\left(\frac{2}{5}\right) - \frac{3}{5}\log_2\left(\frac{3}{5}\right)$$

$$\text{Entropy}(D) = 0.971$$

③ Information gain

Information gain = Entropy(D) − Weighted Average entropy

1, 2, 3 & 4 are 4 possible values.

$\text{Entropy}(D_1) = 0$

$\text{Entropy}(D_2) = -0 - \frac{2}{2}\log\left(\frac{2}{2}\right) = 1$

$\text{Entropy}(D_3) = 0$

$\text{Entropy}(D_4) = 0$

Date :

$$\text{Gain}(\text{Study hours}) = 0.971 - \left[\frac{1}{5}\times 0 + \frac{2}{5}\times 1 + \frac{1}{5}\times 0 + \frac{1}{5}\times 0\right]$$

$$= 0.571$$

Step ④ Calculate split information

SplitInfo(Study hours)

$$\text{SplitInfo}(\text{Study hours}) = \left(-\frac{1}{5}\right)\log_2\left(\frac{1}{5}\right) - \left(\frac{2}{5}\right)\log_2\left(\frac{2}{5}\right) - \left(\frac{1}{5}\right)\log_2\left(\frac{1}{5}\right) - \left(\frac{1}{5}\right)\log_2\left(\frac{1}{5}\right)$$

$$= 1.921$$

⑤ $\text{Gain ratio}(\text{Study hours}) = \frac{0.571}{1.921}$

$$= 0.297$$

Date :

(iii) **Gini index** :- It is a measure of impurity or uncertainty in a dataset. It is used to evaluate quality of a particular split in a decision tree.

CART uses Gini method to create split points.

$$Gini(D) = 1 - \sum_{i=1}^{m} p_i^2$$

p_i is probability that a tuple in D belongs to class C_i

* If a binary split on attribute A partitions data D into D1 & D2, Gini index of D is:-

$$Gini_A(D) = \frac{|D_1|}{|D|} Gini(D_1) + \frac{|D_2|}{|D|} Gini(D_2)$$

$$\Delta Gini(A) = Gini(D) - Gini_A(D)$$

Attribute with lowest Gini index is selected

Date :

Gini Index :

Step 1 :- Dataset

Fruit	Color	Class
Fruit 1	Red	Apple
Fruit 2	Yellow	Banana
Fruit 3	Red	Apple
Fruit 4	Yellow	Banana

Step 2 :- Calculate Gini Index (Gini(D))

$$Gini(D) = 1 - \sum_{i=1}^{c} (p_i)^2$$

c → no. of classes

p_i → probability of a data point in D belonging to class C_i

In this dataset, there are 2 classes :- "Apple" & "Banana".

$$p_{Apple} = \frac{2}{4} \qquad p_{Banana} = \frac{2}{4}$$

$$Gini(D) = 1 - \left[\left(\frac{2}{4}\right)^2 + \left(\frac{2}{4}\right)^2\right]$$

$$Gini(D) = 1 - \frac{1}{2} = \frac{1}{2}$$

Date :

Step 3 :- Calculate Gini Index for attribute "Color" GiniA(D) which has 2 possible outcomes "Red" & "Yellow"

For color = Red, (D1)

$p_{Apple} = \frac{2}{2}$ (2 apples out of red fruits)

$p_{banana} = \frac{0}{2}$ (0 bananas out of 2 red fruits)

Calculate Gini(D1) :-

$$Gini(D1) = 1 - \left[\left(\frac{2}{2}\right)^2 + \left(\frac{0}{2}\right)^2\right]$$

$$Gini(D1) = 1 - (1+0)$$

$$Gini(D1) = 0$$

For color = Yellow, (D_2)

$p_{Apple} = \frac{0}{2}$ (0 apples out of 2 yellow fruits)

$p_{banana} = \frac{2}{2}$ (2 bananas out of 2 yellow fruits)

$$Gini(D2) = 1 - \left[\left(\frac{0}{2}\right)^2 + \left(\frac{2}{2}\right)^2\right]$$

$$Gini(D_2) = 1 - (0+1)$$

$$= 0$$

Date :

Step 4 :- Calculate $\Delta Gini(Color)$

$$\Delta Gini(Color) = Gini(D) - \sum_{j=1}^{2} \left(\frac{|D_j|}{|D|}\right) . Gini(D_j)$$

where,

$|D_j|$ is size of subset D_j created by split

$|D|$ is size of original dataset D

$Gini(D_j)$ is Gini index for subset D_j calculated in step 3

$$\Delta Gini(Color) = \frac{1}{2} - \left[\left(0 \times \frac{2}{4}\right) + \left(\frac{2}{4} \times 0\right)\right]$$

$$\Delta Gini(Color) = \frac{1}{2}$$

Step 5 :- Select splitting attribute.

Color has lowest $\Delta Gini$ value of $\frac{1}{2}$ which is the best attribute to split the data.

Date :

ID3 algorithm:

Algorithm:

1. Start with original dataset that contains all the instances & their corresponding attribute values.
2. If all instances in the current node belong to the same class, then create a leaf node for that class & stop splitting process for this branch.
3. If the attribute set is empty, then create a leaf node for the most common class & stop.
4. Otherwise, calculate information gain for each attribute in the attribute set. to determine the attribute that provides the most useful splitting.
5. Select the attribute with highest information gain as the splitting attribute for the current node.
6. Split the current node into child nodes based on values of selected attribute
7. For each child node created in previous step repeat the process recursively using the subset of instances that correspond to the attribute value of that child node. This process is continued until the stopping criteria (2 & 3) are met for each branch.

Advantages of ID3 algorithm:

1. Simplicity 2. Handles categorical data
3. Interpretability 4. Feature selection

Limitations of ID3 algorithm:

1. Handles only categorical data: Binning & discretization are required for numerical data
2. Overfitting
3. Biased towards attributes with many values
4. Lack of pruning to prevent overfitting

Date :

C4.5 algorithm:

* It is an extension of ID3 algorithm.
* Developed by Ross Quinlan.
* C4.5 algorithm uses a top-down approach to construct a decision tree.

Key improvements of C4.5 over ID3

① Handling missing data:

Age	Income	Will Buy
25	30000	Yes
35	50000	No
45		Yes
30	40000	No

When building decision tree using C4.5, the gain ratio for splitting based on Income would be calculated based only on available records (1st, 2nd & 4th entries). If a new record with a missing income value needs to be classified, C4.5 can predict income value for that item based on known attribute values of other records in dataset.

Date :

② Continuous data:

Square footage	Price
1000	100
1500	150
2000	200
2500	250
3000	300

To handle continuous square footage attribute, C4.5 divides data in to ranges based on attribute values found in training sample.

0 - 1500
1500 - 2500
2500 - 3500

} Suppose this range is used for square footage,

The resulting decision tree might look like this:

If square footage <= 1500, price = 100
" " > 1500, & square footage <= 2500, price = 150
If square footage > 2500, price = 250

By this way the algorithm works on continuous data.

Date :

③ Pruning :

① Subtree replacement :- This strategy involves replacing a subtree with a leaf node if the replacement results in an error rate close to that of the original tree. The replacement process works from the bottom of tree up to the root.

② Subtree raising :- A subtree is replaced by its most used subtree, raising it from its current location to a higher node in the tree. The increase in error rate for this replacement must be determined.

Ex :

Suppose we have a dataset for predicting whether a customer will purchase a product based on their age & income level.

If age < 30, & income = high, then purchase
" " = low, " no "
" >= 30 & " = high, " purchase
" >= 30 & " = low, " no purchase

Validation set → no accuracy

Removing age < 30 since it only applies to a small subset of data.

Date :

After pruning,

if income = high, then purchase
" " = low, then no purchase

④ Pruning :

C4.5 :- It improves upon the older ID3 algorithm by

① Handling continuous data
② Managing missing values
③ Using gain ratio (not just info gain)
④ Pruning to prevent overfitting
⑤ Converting trees to rules for better interpretability.

Ex :- Predicting whether students pass or fail based on study hours, attendance percentage.

60 → pass 40 → fail.

How C4.5 builds tree ?

① Calculate entropy of whole dataset.
How impure or mixed class labels are

$$Entropy(S) = -(0.6 \log_2 0.6 + 0.4 \log_2 0.4) = 0.971$$

② Split Using an attribute :- Study hours.

Subset A : Study hours < 5
" B : " " ≥ 5

Date :

Assume :

Subset A → 20 passed, 30 failed

" B → 40 passed, 10 failed

Entropy(A) = 0.971 Entropy(B) = 0.722

Step 3 : Info gain = $0.971 - \left(\frac{50}{100} \times 0.971 + \frac{50}{100} \times 0.722\right)$

$= 0.124$

Step 4 : Gain ratio = $\frac{\text{Gain}}{\text{Split Ratio}}$ → entropy of how data is split

Gain ratio (Study hrs) = 0.6

" (Attendance %.) = 0.8

Attendance %. is chosen as first split because it has higher gain ratio.

Step 5 :- Pruning tree → Perfect on training data but poor on new (validation data)

Example overfit rule :-

If study hours ≥ 5 & attendance ≥ 80% & has a tutor & drinks green tea → Pass

This is too specific.

Pruned version :- If attendance ≥ 80%, Pass.

Date :

This simpler rule performs better on new data.

C4.5 prunes by

① Removing branches that don't help performance on validation data.

② Replacing subtrees with leaf nodes if it improves generalizations.

Step 6 :- Generating rules :

If age < 30 & income = high → Purchase

" age < 30 " " = low → no purchase

" age ≥ 30 & " = high → purchase

" age ≥ 30 " " = low → no purchase

Rules from tree :

① If age < 30 & income = high → purchase

② " " ≥ 30 & " = low → no purchase

Simplified rule after pruning :

If income = high → purchase

or " = low → no purchase

This is less likely to overfit.

Date :

Concept	Meaning
Entropy	Measure of impurity
Info gain	how much an attribute reduces entropy
Gain ratio	Gain adjusted for how data is split
Pruning	Simplifying tree to avoid overfitting
Rules	If-then statements extracted from tree paths

Split info measures how broadly & evenly data is divided by attribute A.

Date :

CART (Classification & Regression tree)

* CART is a recursive partitioning algorithm that recursively splits the dataset into subsets based on the values of i/p variables.

*. Purpose of CART algorithm is to create a predictive model that can be used for both classification & regression tasks.

* This algorithm uses a top-down greedy approach to recursively split dataset based on the feature that provides best split, determined by Gini impurity criterion for classification tasks or reduction in variance for regression tasks.

* The process is continued till a stopping criterion is met. Stopping criteria → Reaching maximum tree depth, nodes having minimum no. of samples.

Date

how it works?

(1) Select best split :- Calculate the Gini impurity for each feature & select the split that maximizes the homogeneity of resulting subsets (Classification)

$$Gini = 1 - \sum_{i=1}^{n} (p_i)^2$$

p_i → probability of an object being classified to a particular class.

(2) Recursively partition data :- in to 2 subsets. This process is repeated for each subset until a stopping criterion is met.

(3) The result is a binary tree where each non-leaf node represents a decision based on a feature & each leaf node represents predicted o/p.

(4) Handling categorical & numerical features :-
① For categorical features, algorithm considers all possible ways to split data based on categories.
② For numerical features, algorithm considers all possible points to find the best threshold for dividing the data.

(5) The algorithm stops growing tree when stopping criteria is met.
Reaching maximum tree depth,
Having nodes with minimum no. of samples
No further split will improve performance

(6) Pruning strategy is applied to reduce size of tree (reduce no. of nodes) & prevent overfitting.

(7) Accuracy :- leaf node → predicted class label.

Example : CART

(1) Start with original dataset as root node of tree. The original dataset contains
(2) instances of weather data & their corresponding activities.

(3) If all the instances in the current node have same activity (Yes or No), create a leaf node with the corresponding label & stop.

Date :

Date :

(3) If the attribute set isn't empty, Calculate Gini impurity for each attribute (outlook, temp^r, humidity, windy) to determine the attribute that provides most useful splitting

(4) Select the attribute with lowest Gini impurity as splitting attribute

(5) Split current node into child nodes based on values of selected attribute.
Ex:- Create child nodes for each possible value of outlook attribute (Ex:- Sunny, Overcast, rainy).

(6) Recur on each child node, using subset of instances corresponding to attribute value
Ex:- For each child node (sunny, overcast, rainy), repeat the process using subset of instances corresponding to outlook attribute value of that node

(7) Continue the process of calculating Gini impurity, selecting splitting attributes & creating child nodes until the tree is fully constructed

Date :

(8) Pruning (Optional):- After tree is constructed, pruning techniques can be applied to reduce overfitting & improve generalizations.

(9) Prediction:-