UNIT-III

(13)

- Learning with Trees -
- Decision Trees -
- Constructing Decision Trees -
- Classification and Regression Trees -
- Ensemble learning -
- Boosting -
- Bagging -
- Different ways to combine classifiers -
- Basic statistics -
- Gaussian Mixture Models -
- NearestNeighbor Methods -
- Unsupervised learning
- K means Algorithms

* Learning with Trees

→ Learning with Trees is a method of ML.

→ A method where decisions are made by asking a series of questions.

→ We build a Decision Tree
- that splits data based on features until we reach a prediction.

→ It is a method in ML that uses decision trees to make predictions.

→ Decision tree is built from training data.

→ the decision tree is like a flowchart of decisions.

→ Each internal node represents a decision on a feature. * internal node = decision

→ Each branch represents the result of the decision.

* branch = Outcome of the test

→ Each leaf node gives the final decision (eg: Pass/Fail)

* leaf node = final decision

→ Tree is built by splitting data based on features.

→ It is mainly used for classification tasks

→ But can also be used for regression

→ It is a type of Supervised learning.

→ It builds a model in the form of a tree structure.

→ The model predicts a target
- by learning decision rules from input features

→ Learning with trees is a ML technique
- for classification and regression.

→ It uses a tree-like structure to make decisions.

→ It is a form of supervised learning.

→ Final decisions are made at the leaf nodes of the tree.

* Decision Tree

→ A Decision Tree is a tree structured model used to
→ It is a flow chart like model / tree.
→ It helps in decision making.
→ By splitting data, it makes decisions.
→ It is used for both classification and regression.
→ It splits the data into branches based on i/p.
→ Each internal node represents a decision
→ Each branch represents result.
→ Each leaf node represents final output.
→ It follows if-else type conditions to predict output.

Parts / Components of a Decision Tree

1. Root Node – top node
2. Internal Node – decision point
3. Leaf Node – Represents the output
4. Branches – Shows outcome (or) final output
5. Path – A route from root to leaf node.

Decision Tree Algorithms

→ ID3
→ CART
→ Random Forest
→ C4.5

→ It asks a series of Yes/No questions.
→ Based on the answers, it moves down the tree.
→ Finally, it gives a result at the leaf node.

Advantages

- Easy to understand
- Easy to use
- works for both numbers & categories
- No need for scaling or normalization

Disadvantages

→ May overfit
→ Can change a lot with small changes in data

Applications

- Student result prediction
- Loan approval
- Medical diagnosis
- Weather prediction

ex: If age > 18 → "Adult"
else → "Child"

Real-world use

→ Loan Approval
→ Medical diagnosis

* Constructing Decision Trees

→ Constructing Decision trees means building the tree from data
- by choosing the best features at each step.

→ It is the process of building a tree
- that predicts outcomes.

Goal ⇒ To split into small, pure subsets
⇒ Divide the data into pure subsets.
⇒ Build the tree, it makes accurate predictions

Steps to construct a Decision Tree

1. Start with the full dataset.
2. Calculate entropy of the dataset.
3. For each feature:
   - Split data based on the feature
   - calculate Information Gain.
4. Choose the feature with the highest Information Gain.
5. Make that feature as decision node.
6. Split dataset based on that feature.
7. Repeat the process for each child node.
8. Stop when:
   - All samples are of one class
   - No features are left
   - Tree reaches maximum depth or limit

* Classification and Regression Trees

→ Classification and Regression Trees are short for CART.

→ It is a decision tree algorithm.

→ It is used for both classification & Regression.

→ used to predict a category (class)

Types : Ex: Predict if email is "Spam" or "Not Spam"

1. Classification Tree: Output is discrete.

   eg: Spam or not
   Pass or fail

2. Regression Tree: output is numerical

   eg: age
   Salary
   weight

Splitting Criteria

- Classification tree: Uses Gini Index or Entropy.
- Regression tree: Uses Mean Squared Error (MSE) or Variance.

Steps to build CART

1. Start with the entire dataset.
2. For each feature, calculate the best split points
3. Choose the feature with the best split (lowest Gini or MSE)
4. Split the data into two subsets.
5. Repeat the process.
6. Stop when :
   - A stopping rule is met
   - Node is pure
   - Node can't be split further.

Gini Index

$$* \text{Gini} = 1 - \Sigma(p_i)$$

MSE

$$* \text{MSE} = \frac{1}{n} \Sigma (y_i - \hat{y})^2$$

Advantages

- Easy to understand.
- Works both classification & Regression.
- No need for scaling or normalization.

Disadvantages

- May overfit.
- Sensitive to small changes in data.

Real-world Examples

* Classification Tree:
  - Predict loan approval (Approved/Rejected)
* Regression Tree:
  - Predict house price.

Applications

- Finance
- Healthcare
- Marketing
- Real estate.

* Ensemble Learning = combing multiple models

→ Ensemble learning Combines multiple models.
→ Inorder to create a stronger model, it (to improve accuracy & Performance)
combines multiple models.

→ It improves :
- accuracy
- robustness (strong)
- Performance.

→ It is based on the idea :

Group of weak learners = Strong learner. better decisions

→ one model may make mistakes, but a group can make
→ Single model may be inaccurate & unstable.
→ Combining models reduces bias, variance & overfitting.

Types of Ensemble ~~lea~~ :

1. Bagging - builds multiple models in Parallel,
2. Boosting - builds models sequentially (one after another)
3. Stacking - combines outputs of multiple models

Advantages	Disadvantages
- Higher accuracy	→ More complex (than Single models)
- High Performance	→ Slower
- robustness.	→ need memory & computation

Real-life Examples

- Spam detection
- Fraud detection
- Credit card fraud detection
- Stock price prediction.

Ensemble Learning Algorithms

- Random Forest
- AdaBoost
- XGBoost
- Gradient Boosting.
- Stacking classifier
- Voting classifier.

Why ?

⇒ A single model may be weak.

⇒ Many weak models together = strong model.

⇒ Reduces : - errors
- overfitting
- Variance

* In ML, using multiple classifiers together is called Ensemble learning.

* Boosting

→ Boosting is an ensemble learning method.

→ It combines multiple weak models to create a strong model.

→ Models are trained one after another (Sequentially)

→ Each model tries to fix the errors made by the previous one.

> * Boosting = combining many weak models to make one strong model

Working

1. Train the weak model first
   eg: Small tree.
2. Check which data points were predicted wrong.
3. Gave more importance → weight to wrong points.
4. Train next model.
5. Repeat this process
6. Final Output = Sum of all models.

Types

- AdaBoost
- XGBoost
- Gradient Boosting

Advantages | Disadv / Real-life examples

Same as Ensemble learning.

P

* Bagging

→ It is an ensemble method.

→ Bagging stands for Bootstrap Aggregating.

→ It trains multiple models in parallel.

Algorithm

- Random Forest
- Bagged Decision Trees
- Bagged KNN
- Bagged Logistic Regression
- Bagged Neural N/ws

Working

1. Create multiple bootstrap samples
2. Train a separate model (or) Train model
3. Combine predictions : (or) Predict
   - Majority Voting for classification
   - Average for regression

⇒ Models are trained in parallel.

⇒ Bagging reduces Variance

⇒ It avoids overfitting.

Eg:

Example

- Dataset: 5 Training Samples → A, B, C, D, E

Step-1: Create 3 bootstrap samples
- with replacement

- Sample 1 → A, B, A, D, E
- Sample 2 → B, C, C, D, A
- Sample 3 → D, E, E, A, C

Step-2: Train 3 decision trees

Step-3: Make Prediction on a new input

- Tree 1 → YES
- Tree 2 → NO
- Tree 3 → YES

- Final output: Majority Vote = Yes

* Different ways to combine classifiers

Techniques to combine classifiers :=

1. Majority Voting
2. Averaging
3. ~~Stak~~ Stacking
4. Boosting
5. Bagging
6. Blending
7. Weighted Voting.

1. Majority Voting

→ Used for classification problems.
→ Each model votes for a class.
→ The class with most votes is selected.

Example: 3 classifiers → YES, YES, NO

Output = YES

2. Averaging

→ Used for Regression problems.
→ Final Prediction = average of all model outputs
→ Reduces variance
- & Smoothens predictions

3. Weighted Voting

→ Models are assigned with weights
- based on accuracy or confidence.

→ Final prediction = Sum of weighted outputs.

4. Boosting

→ Models are trained sequentially.
→ Each model corrects the mistakes of the previous one.
→ Example algorithms: AdaBoost, Gradient Boosting
→ Final output: Weighted sum of all models

5. Bagging

→ Models are trained in parallel.
→ outputs are combined using voting or averaging.
→ Reduces overfitting & variance.
→ Ex: Random Forest

* Basic Statistics

→ Statistics is the branch of mathematics.
→ It deals with collecting, analyzing numerical data.
→ It is study of collecting, organizing and analyzing data.
→ It helps in understanding patterns
→ It helps in:- understanding data.
- Making predictions
- Selecting features
- Evaluating model performance
→ calculating

Types

1. Descriptive statistics – Summarizes data.
2. Inferential statistics – Makes predictions.

Basic Statistical Terms

1. Mean (Average)
→ Sum of all values divided by no. of values
→ Formula: Mean = $\frac{x_1 + x_2 + \dots + x_n}{n}$
ex: 5, 7, 9, 11, 13 ⇒ Mean = 9

2. Median
→ the middle value when data is sorted.
→ If even no. of values → average of middle two.
ex: 5, 7, 9, 11, 13 ⇒ Median = 9

3. Mode
→ the value that occurs most frequently.
ex: 5, 7, 9, 11, 13 ⇒ No mode

4. Range

→ Difference b/w maximum & minimum value

ex: 5, 7, 9, 11, 13 ⇒ Range = 13 − 5 = 8

5. Correlation.

→ Shows the relationship b/w two variables

→ Ranges from −1 to +1.

+1 → Strong +ve relation

0 → no relation

−1 → Strong −ve relation

6. Variance

→ Measures how much data points differ from the mean.

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

7. Standard Deviation (SD)

→ Square root of variance.

→ Shows spread of data.

8. Skewness

→ Measures asymmetry of data

+ Skewed left (−ve)

− Skewed right (+ve)

* Gaussian Mixture Models

→ Gaussian Mixture Models is short for GMM.

→ GMM is a probabilistic model.

→ It is used for clustering.

→ Each cluster is represented by a Gaussian curve.

→ It assumes data is generated from a mixture of several Gaussian distributions.

→ It provides soft clustering.

→ It is more flexible than K-means.

→ It find hidden patterns in data.

Key Concepts

1. Gaussian Distribution

→ Also called Normal Distribution.

→ Bell-shaped curve

2. Parameters in GMM

→ Mean (μ)

→ Covariance (Σ)

→ σ

*Nearest Neighbour Methods

→ these are supervised learning algorithms.

→ used for classification & regression.

→ The output is based on the closest (neighbor) data points.

⇒ when given a new input, the model finds the nearest data points

⇒ then make predictions.

→ Popular Algorithm is K-Nearest Neighbors (KNN)

KNN

→ KNN is lazy learning or ~~eager~~ learned

→ no training step/phase

Working

⇒ Choose the value of K.

⇒ Calculate the distance
- b/w the new point & all points

⇒ Select K nearest points.

⇒ For Classification: Take Majority vote

⇒ For Regression: Take average

Example

* Training data: (2,3) → class A
(5,4) → class B
(1,1) → class A

* New point: (3,3)

* K=3 ⇒ closest = (2,3), (1,1), (5,4)

* votes: A, A, B → class = A

Advantages

→ Very simple
→ easy to understand
→ No training
→ works well with small datasets.

Disadvantages

→ Slow with large datasets.
→ Need to choose a good value of K.

Applications

→ Fraud detection.
→ Handwriting recognition.
→ Recommendation Systems
→ Face recognition

Distance Metrics Used

- Euclidean Distance
- Manhattan Distance
- Minkowski Distance

⇒ KNN is a supervised ~~lea~~ algorithm.
⇒ It is used for classification & regression.
⇒ It stores the whole training data
- & makes predictions based on closest data points.

⇒ Distance is measured using ~~Eud~~ Euclidean, Manhattan or other formulas.
⇒ For classification, majority class

Ex: Predicting fruit type based on color, size

Real-world Use:

- ~~Recom~~

NearestNeighbors Methods

- KNN
- Weighted KNN (Neighbors closer to the point have more weight)
- Radius Neighbors (considers all neighbors within a fixed radius)
- Ball Tree / KD Tree (Faster search)
- Approximate NN (Fast but gives approximate nearest points)