ML
(V-4)

KNN -

- K-Nearest Neighbour
- Supervised learning approach.
- assumes similarity b/w new case/data & available test case & put new case into category that is most similar to categories available.
- Regression & classification (more preffered)
- Lazy learner algo → not learning from training set immediately
  → stores dataset & time of classification → action.

Adv → simple to implement
robust → noisy training data
more effective → training data → large

Disadv → • need to determine K value
• computation cost high.

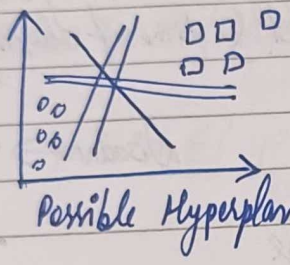
Working →
- select no. of K of neighbors
- Calculate Euclidean distance of K no. of neighbors
- Take K nearest neighbors as per calculated Euclidean distance
- K-neighbors, count no. of data point in each category
- Assign new data pain to category which no. neighbors is most.
- Our model is ready.

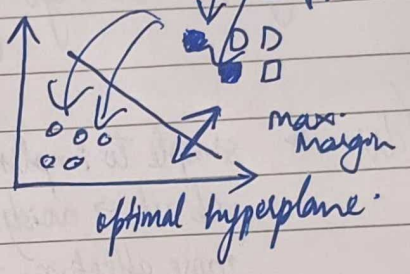
$d=\sqrt{(x_1-x_2)^2+(y_1-y_2)^2}$.

Why KNN → discovery category or class of selected dataset without difficulty

SVM

- Support Vector Machine
- supervised learning
- classification & regression problem
  (more)
- boundary data points to create a hyperplane that differentiate data pts
- support vectors.





Types → Linear SVM → • linearly seperable info.
• data → labelled two classes → unmarried directly line.

Non Linear SVM → • non linearly seperated facts.
• dataset → can't be labbled through usage of directly line
• fact → non-linear info
classier → non-liner SVM classifies.

Hyperplanes → decision boundary that help in classifying datapoints

Supportvectors → datapoints that are closest to hyperplane & influence posn & orientation of hyperplane.

* appln SVM → Image classificn - Handwritten characted recogn

* Limitation → • sensitive to noise.
• Lies in choice of kernel
• optimal design for multiclass SVM classifier

Bagging

Goal → Reduce Variance

Learning → Parallel

more memory use

~~high~~ receives equal weight.

solves overfitting problem.

• used if classifier is unstable

eg → Random Forest

• combine homogeneous weak learners.

• aggregates predictions from weak learners

Boosting

Reduce Bias.

Sequential.

less memory usage

Models are weighted acc to their performance.

only reduce bias

• Used if classifier is stable & simple

eg → AdaBoost, XGBoost

• weak learners ain't independent.

• weights of misclassified are increased.

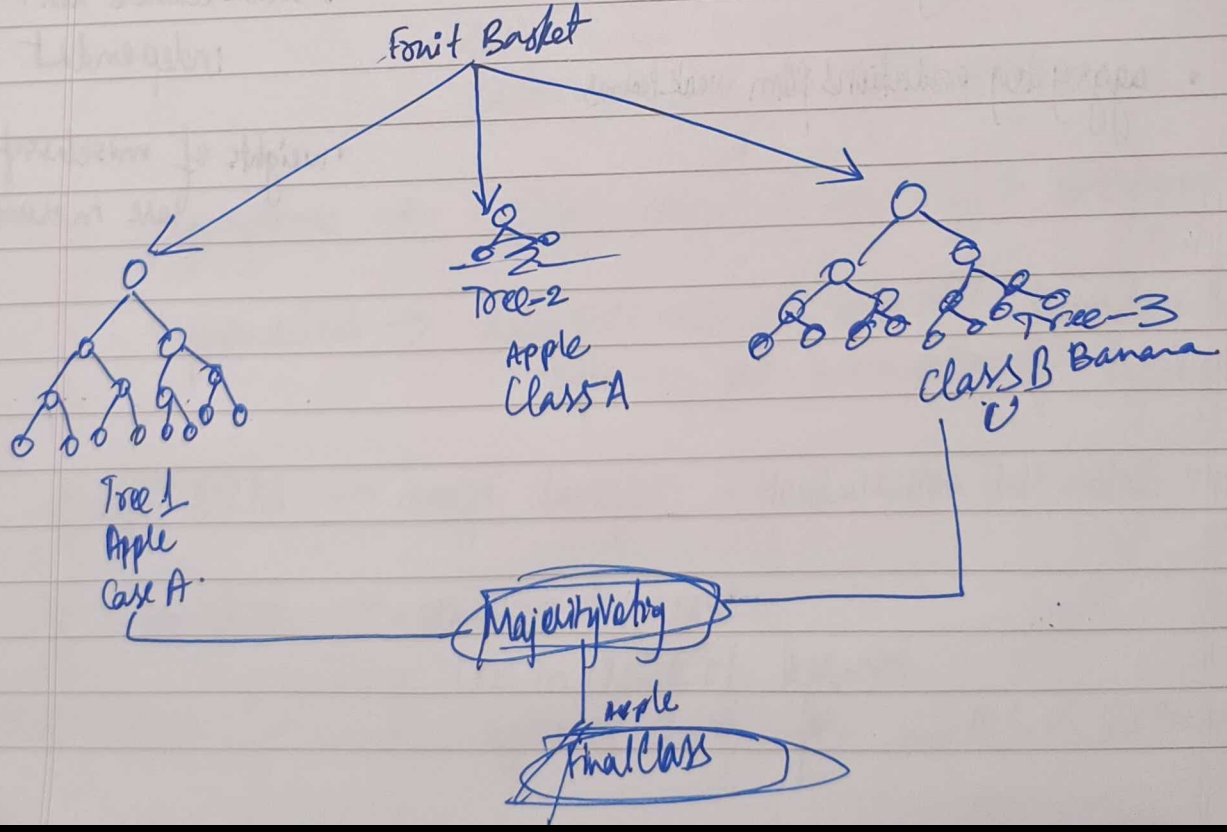
* Random Forest-

→ made up of multiple decision trees.
→ classification & reg" problem.
→ output of multiple decision trees to reach single result

Adv
- less training time
- output with high accuracy → large dataset → efficient
- maintain accuracy → large portion of missing data

Disadv
- not more suitable for regression task

Appln →
- Banking
- Medicine
- Land use
- Marketing.

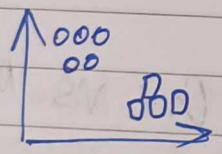


Binary Classfn

- Supervised learning
- It is two types of groups. → max two classes
- eg → Medical Test → disease yes or no

Algo → • Logistic regn
- K-nearest neighbours
- SVM
- Naive Bayes

eg → • spam detection → spam yes or no
- conversion predn → buy or no

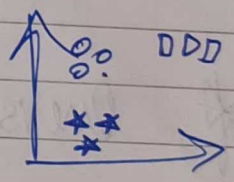


Multiclass Classfn

- Supervised learning
- May be any variety of classes - many
- eg → classify email categories

Algo → • k-Nearest nei
- Decision trees.
- Random forest
- Gradient

eg → • plant species classificat
- optical character recgn



Balanced classification

one class.
- equal to other class

- ML models work best when no. of samples in each class are about equal.

Imbalanced classifical.

- If observations in one class is way higher than other class
- May not be good ml model

~~Techniques~~.

Cause → 1) Sample Errors
2) Problem Domain.

Technique to handle class imbalance-

- Random Resampling —
removing samples from majority class & add more in minority
- Tomek links —
  - Majority class of each pair removed.
  - Link pairs of very close instance but opp classes
- SMOTE (synthetic Minority Oversampling Technique)
  - Generate Synthetic Data → minority class.
  - Uses KNN.
- Class Weights → Biasing any particular class

One vs One

- Heuristic method for binary classification.
- splits multiclassification into Bi-nary problem.

eg → classify colors → R B G.

classifn → Binary Probde-1 → red vs Blue
" " 2 → Red vs Green
" " 3 → Blue vs Green

no. of binary datasets will be created one vs One.

$\frac{C \times (C-1)}{2}$ → C → no. of classes.

argmax sum of score predicted → class label

One vs All.

- Heuristic method for using binary class^n for multiclassification.

eg → R G B V classifn

3 Class Probem 1 → R Vs [G, B, V]
2 → G Vs [R, B, V]
3 → B Vs [R, G, V]
4 → V vs [R G, B].

- one model for one class
- each mode predicts → class membership probability
- argmax score → predict class

Precision → $\frac{TP}{TP+FP}$ Recall = $\frac{TP}{TP+FN}$

* Macro average &
  - when all classes needs to be treated equally

Microaverage
↓
need to weigh each instance of prediction equally.

* Cross Validation -

- given dataset → randomly create training & testing dataset & test out various model to see which works best.

- educate model → Usage of subset of facts set after which compare use of complementary subset of facts set

Two ways → 1) Non exhaustive → don't split dataset in all ways leave subset of data for valn

2) Exhaustive → split data into all possible ways into training & validation sets

Methods

1) Holdout Method
- Non Exhaustive
- Divide data → 70:30 or 80:30 training/testing
- data is shuffled randomly before splitting

2) K-fold cross valn
- split data in k parts
- Train data on (k-1) parts & test data on 1 part.
- perform steps k times.
- Total combo is nCp

3) Leave P out
- exhaustive approach
- take p points from total no. of points.
- Train model on (n-p) points & test on p remaining
- Repeat all possible options
- Avg accuracy → Total p