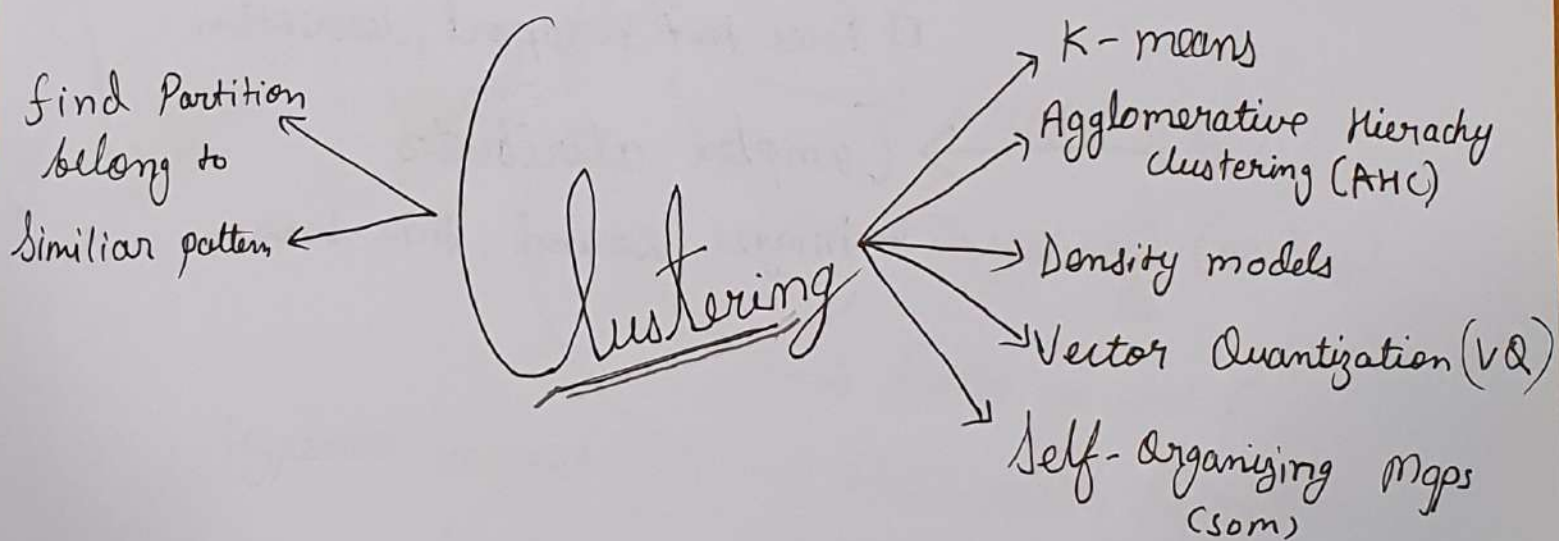
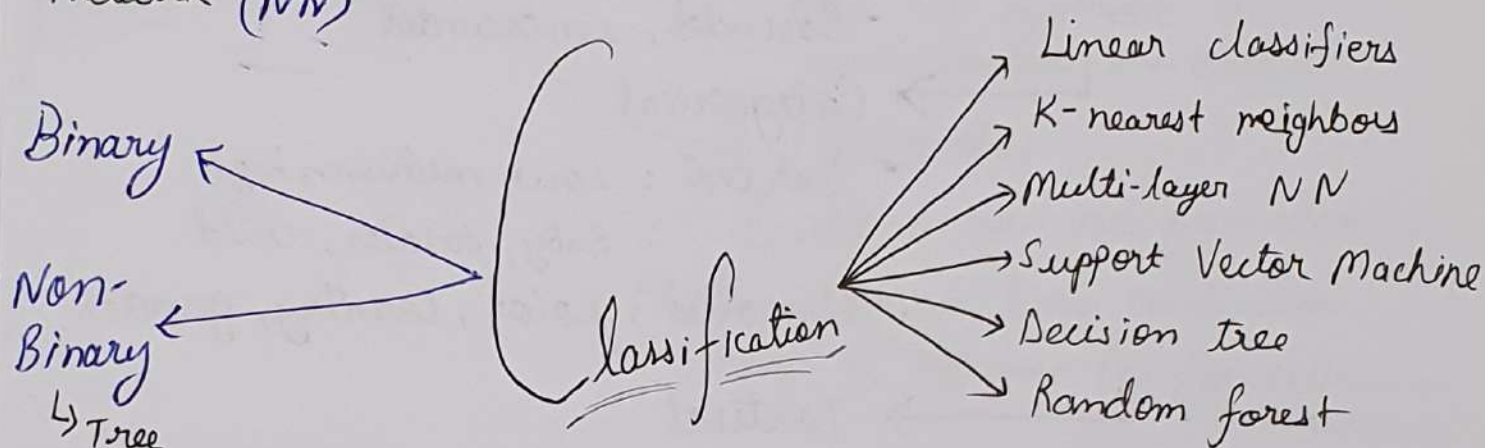
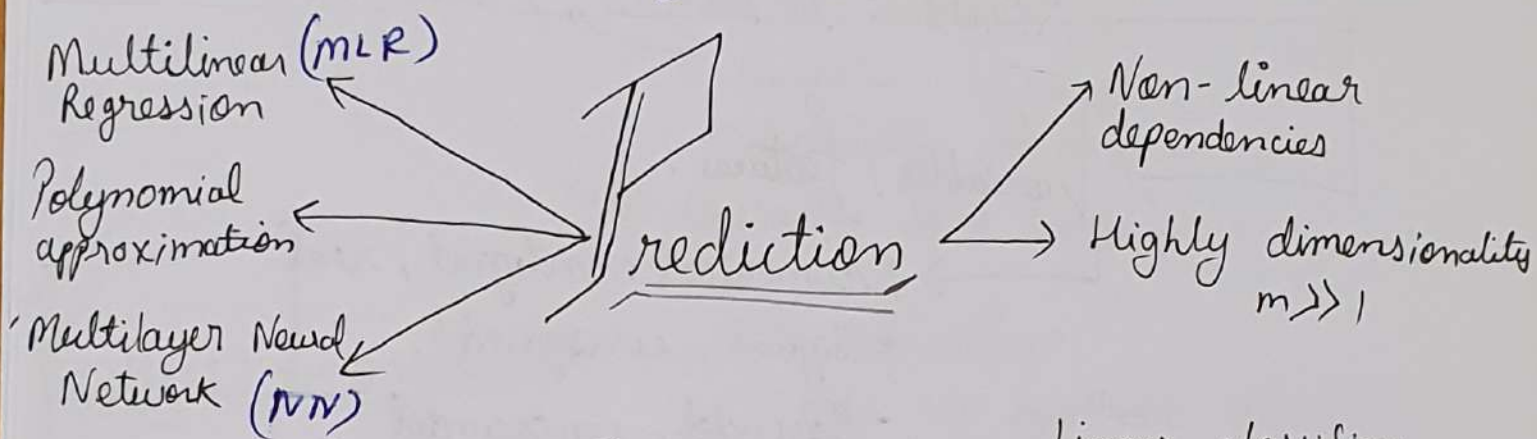
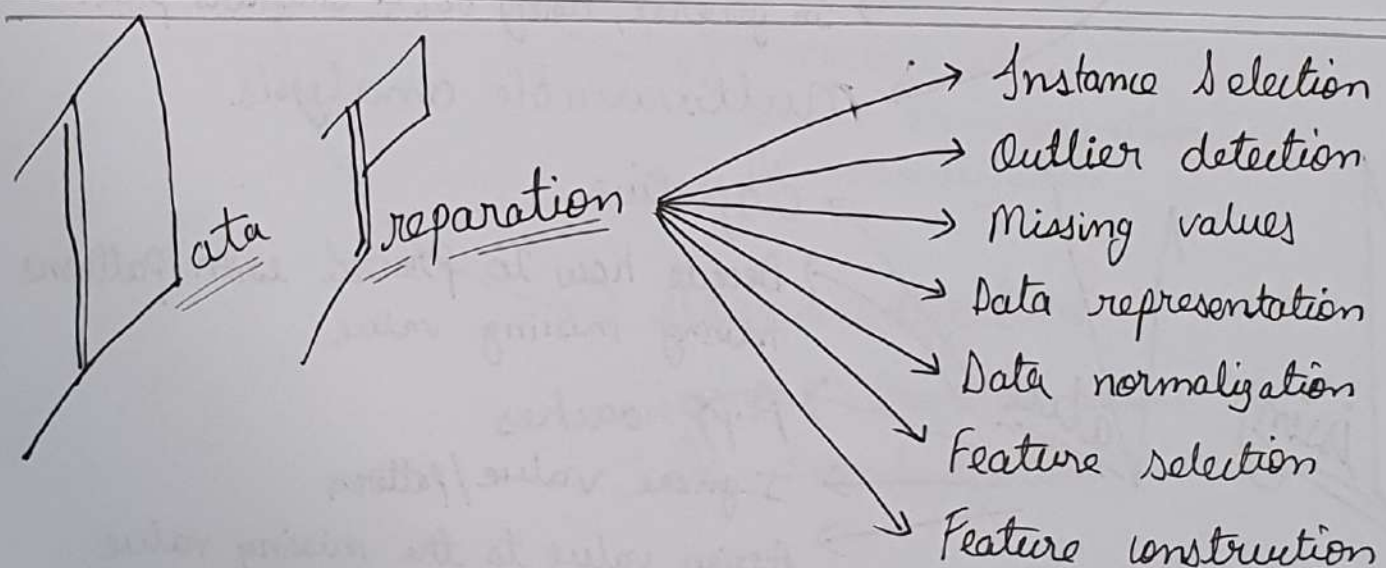
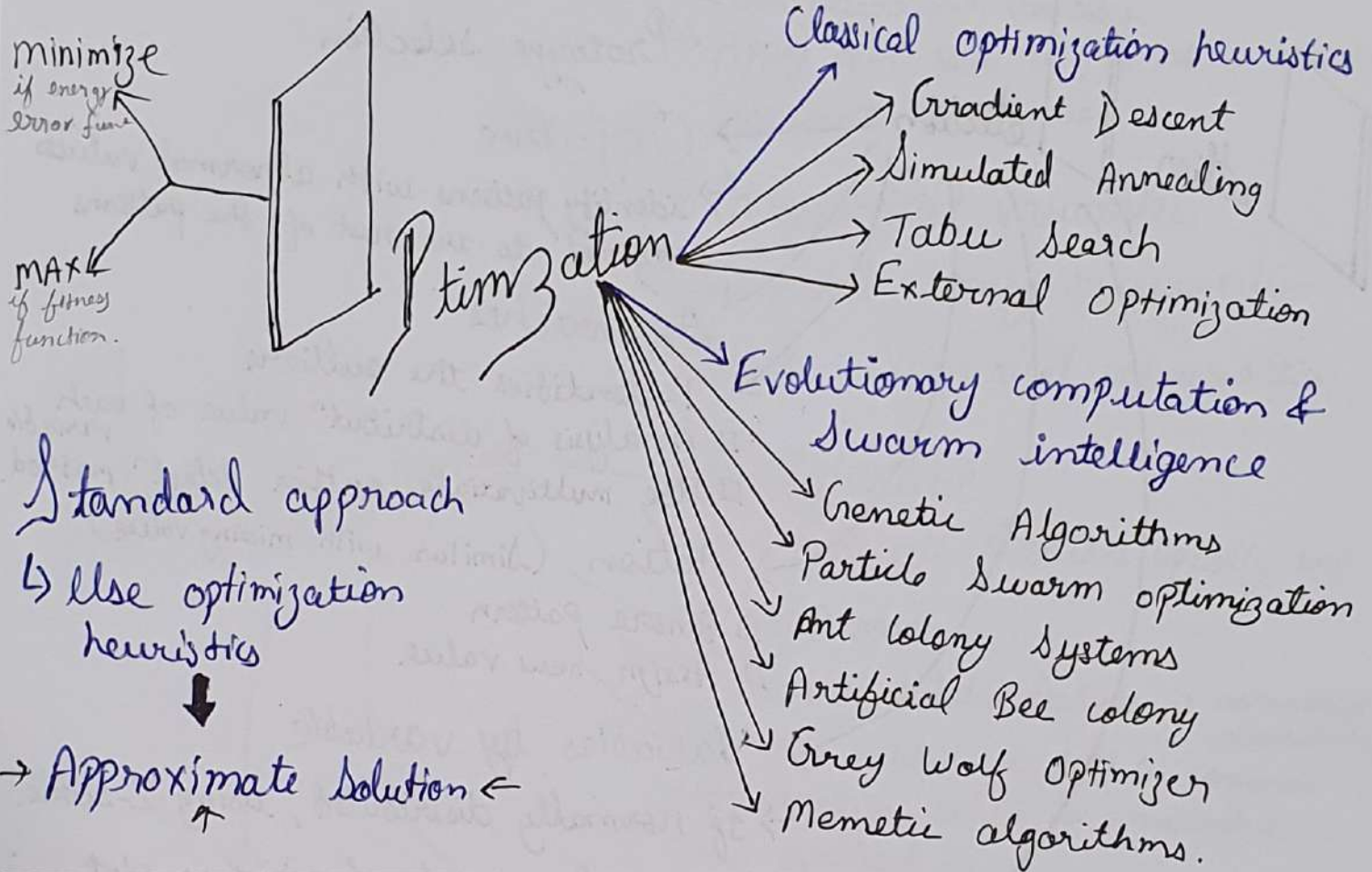
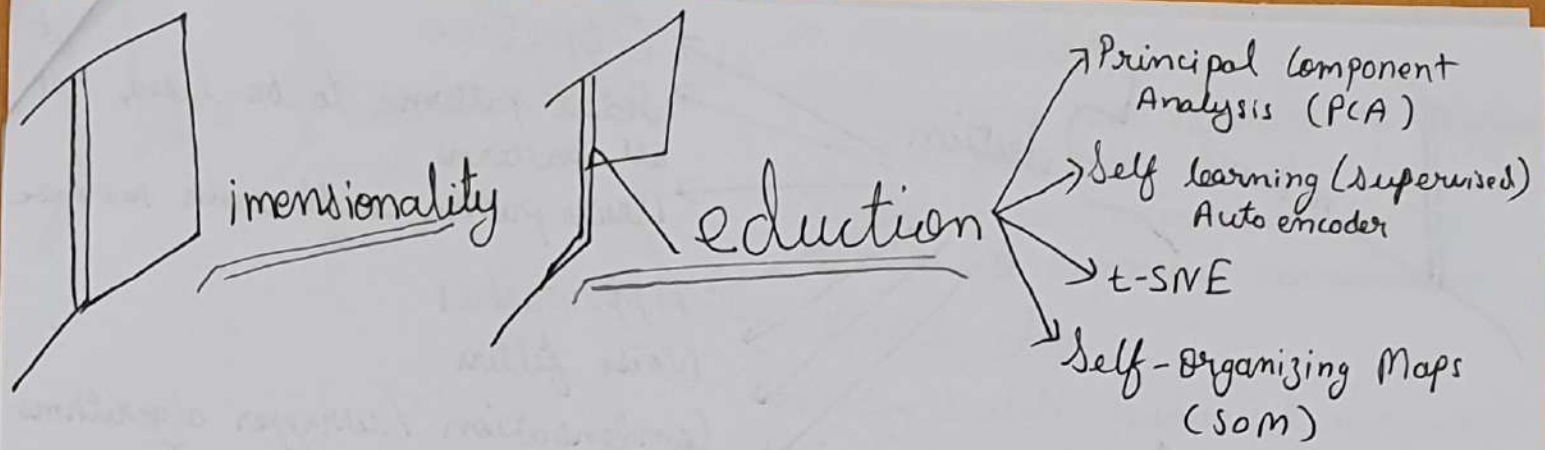


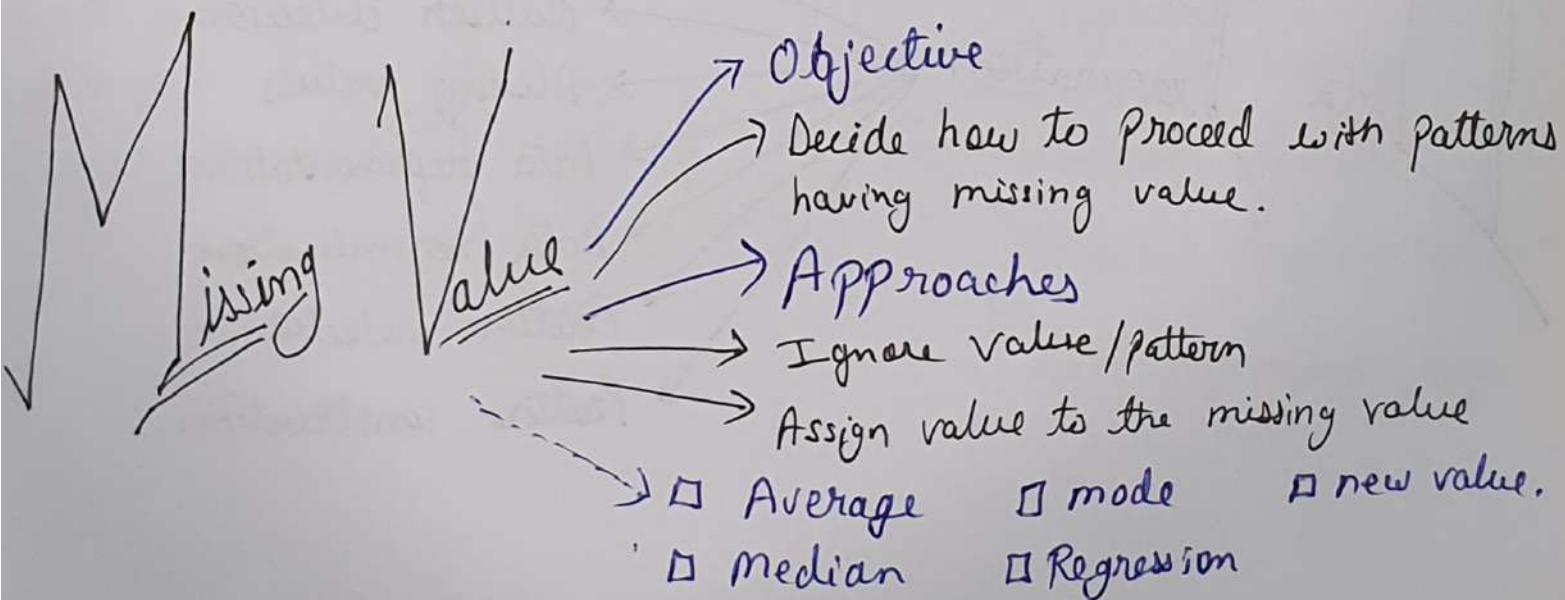
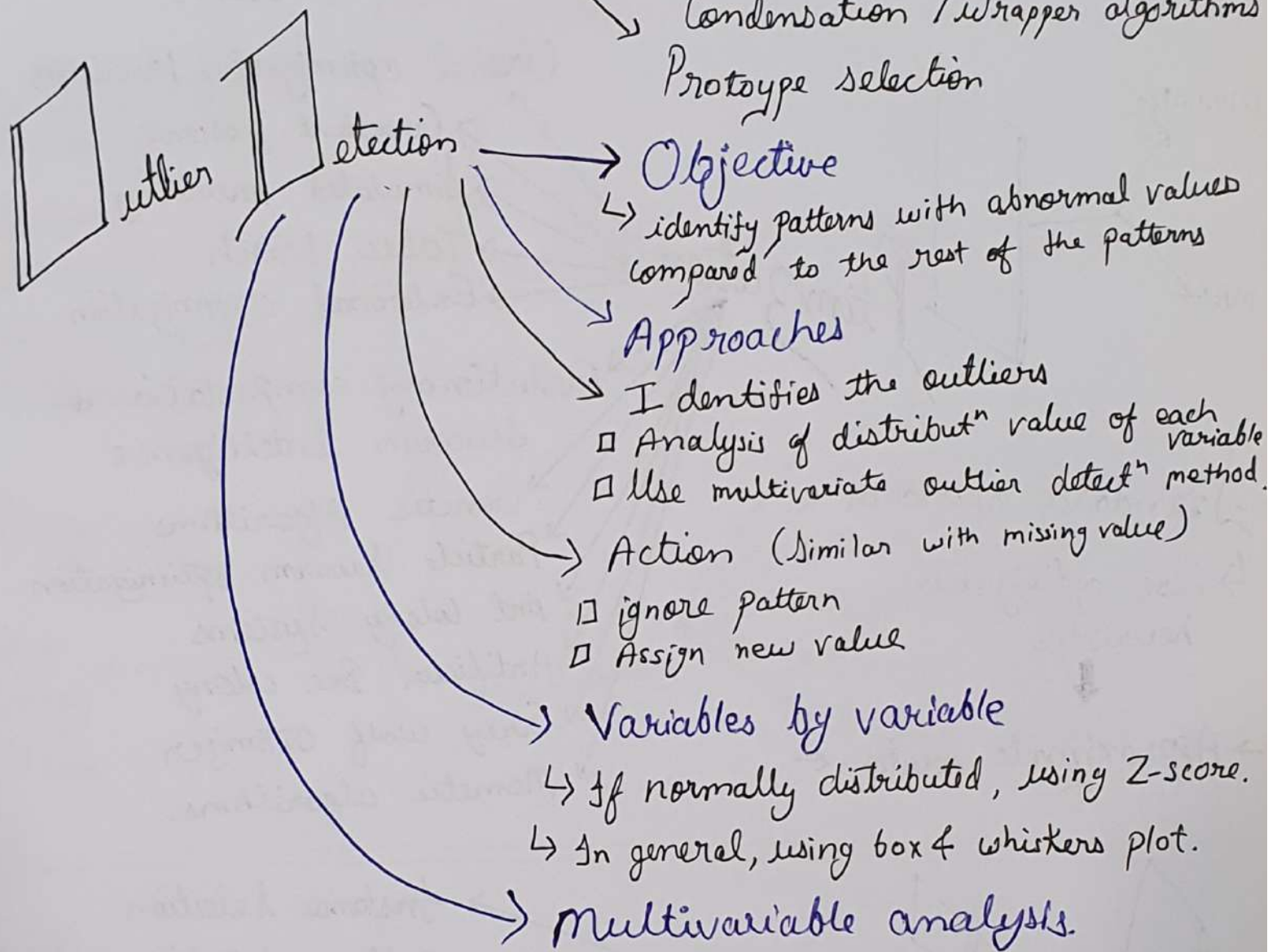
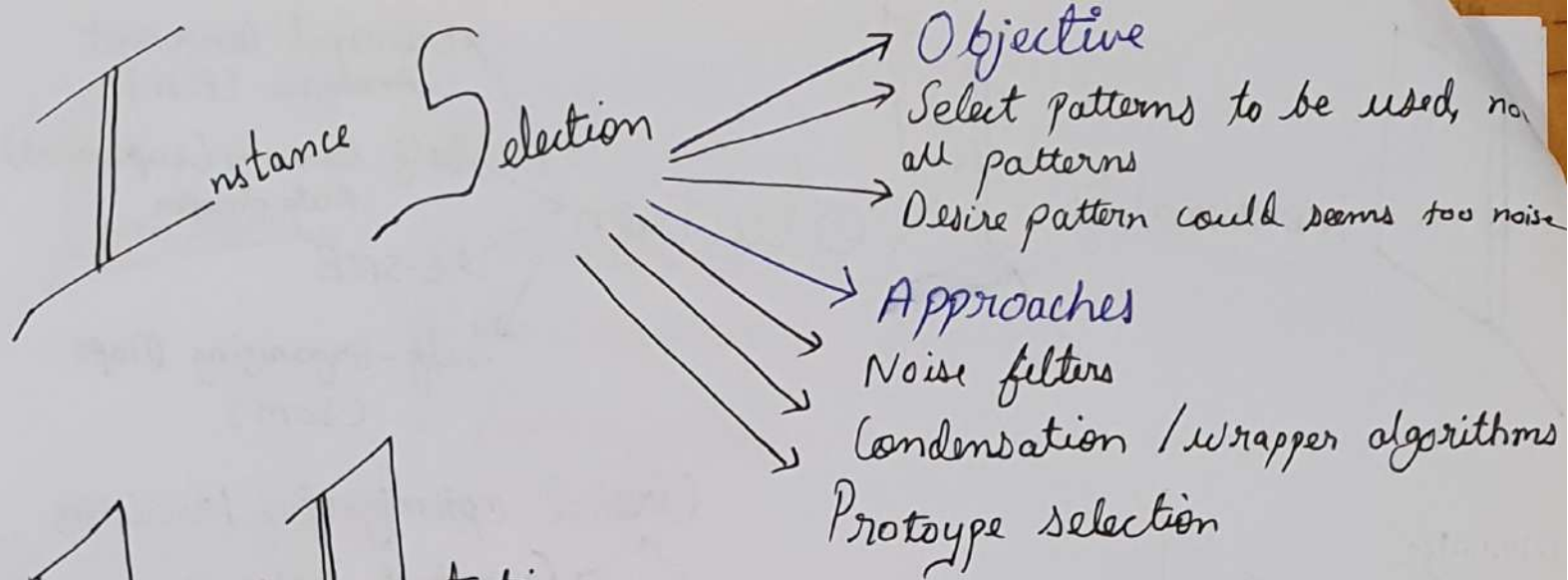
Problems →

- 1) Prediction
- 2) Classification
- 3) Clustering
- 4) Dimensionality Reduction
- 5) Optimization.
- 6) Encoding
- 7) Prototyping
- 8) Visualization
- 9) Familiarity
- 10) Feature mapping











# Data Representation

## Objective

- ↳ Algorithm need number, no categories or free text.
- ↳ Substitute non-numerical value by appropriate numerical representation.

## Approaches

- ↳ Finding suitable numerical representation for categorical variables.
- ↳ Apply NLP tech. to free text to identify relevant information.

## Categorical variables.

↳ Unsorted : Unary representation

↳ Sorted : Numerical representation

# Data Normalization

## Objective

- ↳ Make all features equally imp.

## Approaches.

- ↳ Each unrelated feature is normalized independently
- ↳ For each related var. sets a common normalization could be meaningful

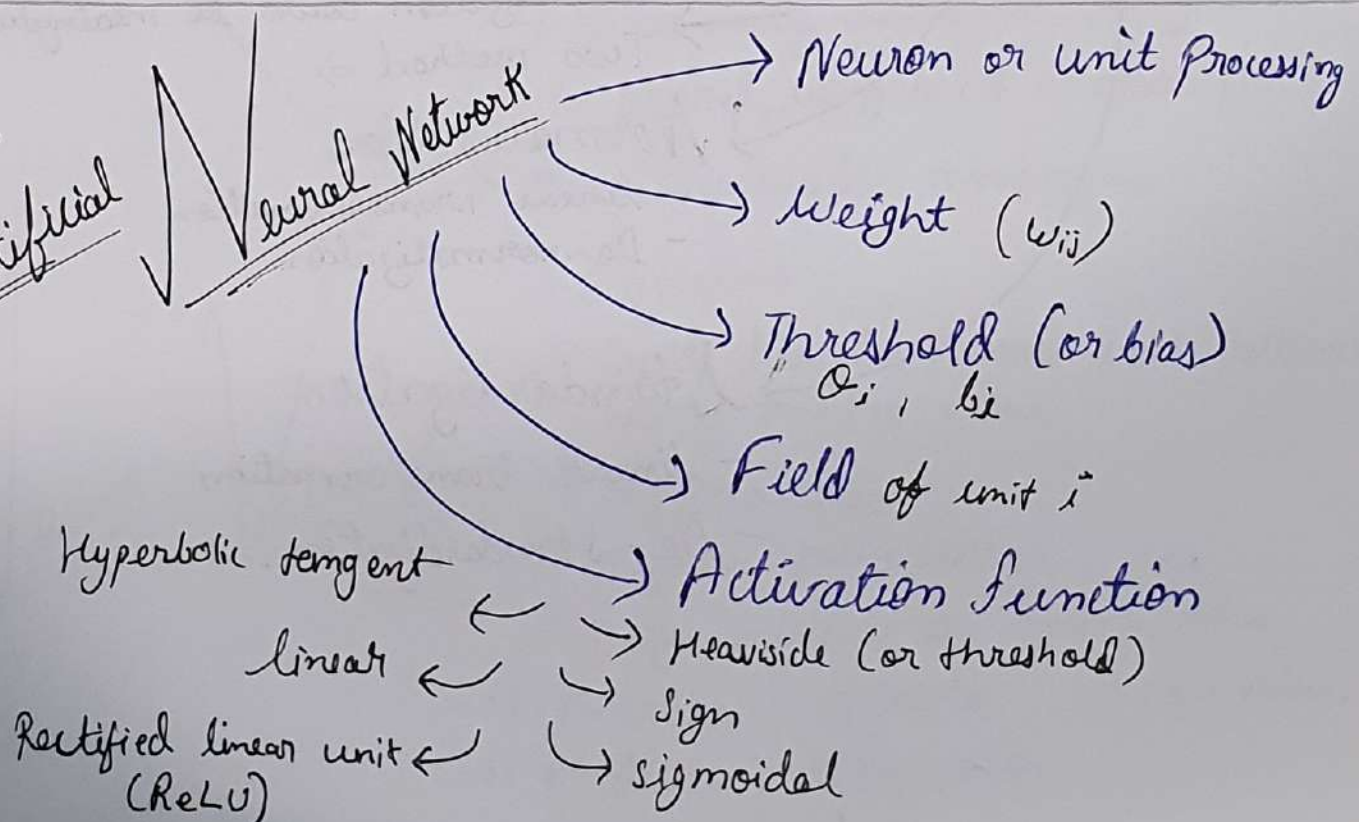
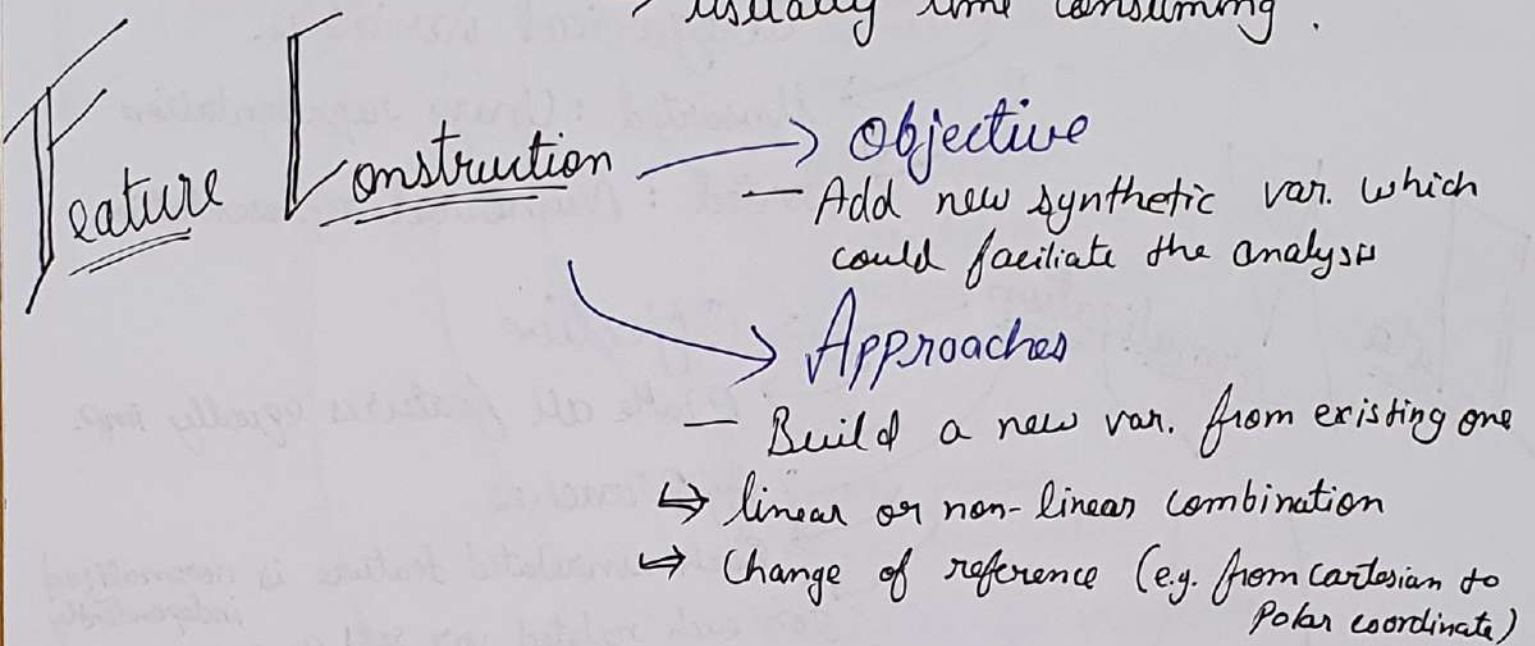
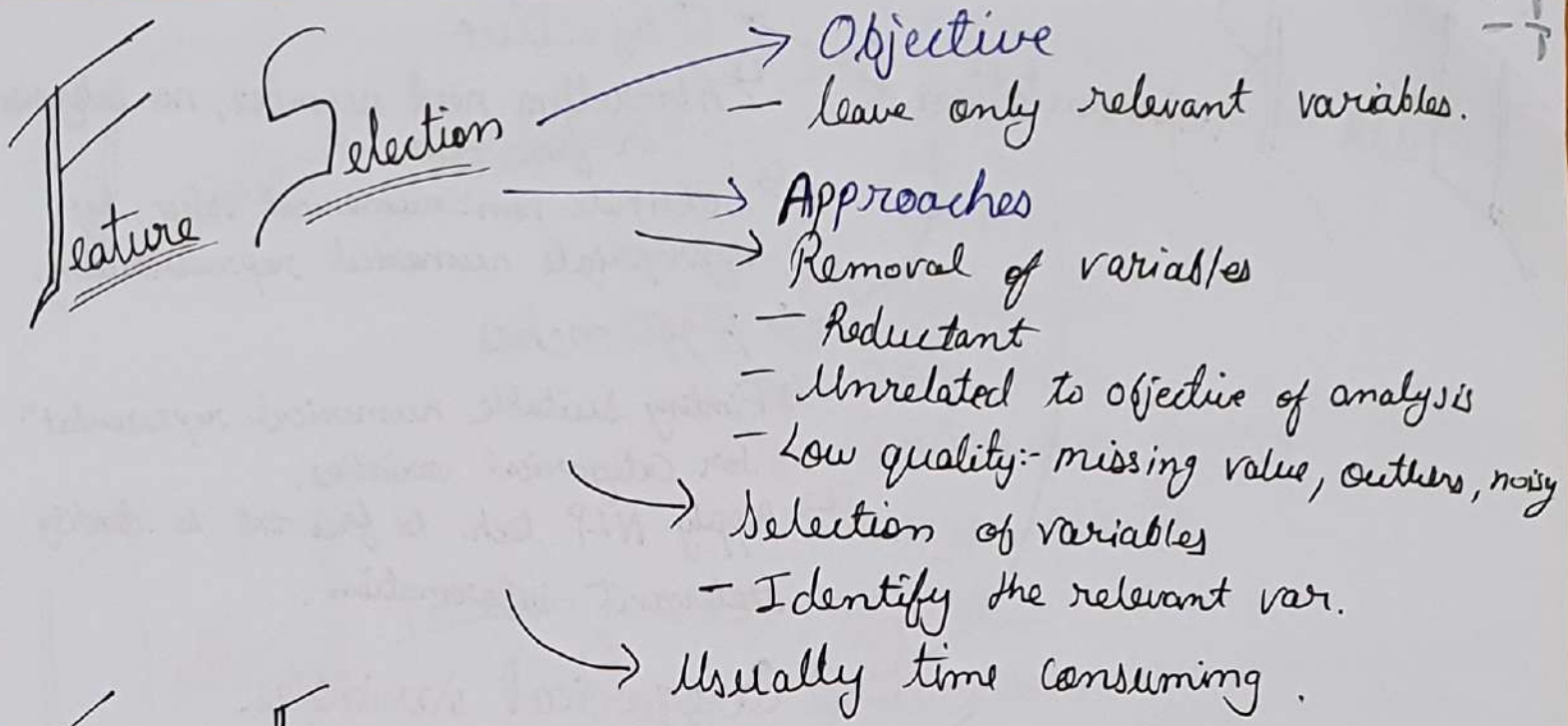
Two method  $\Rightarrow$

## Normalization

- linear transformation
- De-normalization

## Standardization

- linear transformation
- De-standardization.





Supervised Learning

- Prediction
  - give the best prediction to approximate data
  - Multilinear regression (MLR)
- Classification
  - K-nearest neighbors (K-NN)
  - Logistic regression, naive bayes.  
(based on probabilities)

Multilinear Regression (MLR)

- Unidimensional output
- Minimize quadratic error over training patterns
- Results

Logistic Regression

- Statistical model for binary classification from set of input data (continuous/discrete)
- It is based on a logistic function.
- To fit model (identify best co-efficient) we use Maximum Likelihood Estimation (MLE)

Naive Bayes Classifier ⇒ Algo. for classification based on probabilities & prior information.

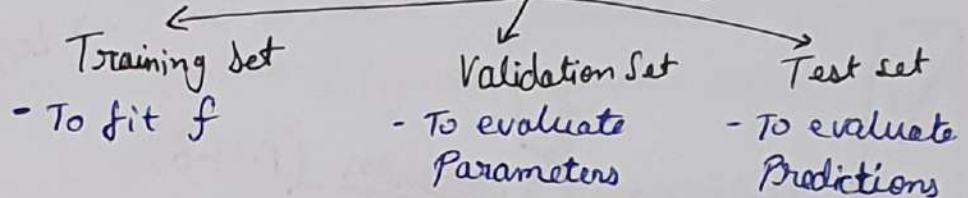
K-Nearest Neighbors (KNN) ⇒ Dataset  
Predict  $x$ , based on its  $K$ -nearest neighbors in dataset

# Validation of Predictions

→ If applied to whole dataset  
↳ They measure quality of fitting not the quality of Prediction

## Solution

### □ Division of dataset



→ Cross-validation (multiple validation)  
- 3-fold cross validation

→ Validation is necessary to avoid overfitting

→ Prediction evaluation measures

- ↳ Mean squared error
- ↳ Mean absolute error
- ↳ Relative absolute error.

→ Evaluation of Prediction ∴ Classification of evaluation measures.

- Sensitivity ✓
- Specificity ✓
- Fall-out ✓
- Precision ✓ (or Rand index)
- Accuracy ✓ (or threat index)
- Jaccard ✓
- F1 score ✓
- ROC-curve → Area under the curve (AUC) ✓
- Confusion matrix (or contingency table) ✓



# Back-Propagation

Application of gradient descent to minimize the quadratic error over the training set of a MLR

## Observations

- ↳ gradient calculation (chain rule of derivative)
- ↳ It is needed a differentiable activation function (Sigmoid, hyperbolic tangent, ReLU)

## Error Back-Propagation

- ↳ Output-layer
- ↳ Rest of the layer (in backwards order)
- ↳ Amount of weights & thresholds update
- ↳ Includes
  - ↳ Learning rate  $\eta$
  - ↳ Momentum  $\alpha$

## Derivative of activation function

- ↳ Sigmoid
- ↳ ReLU

## Comparison

### • Batched back-Propagation

- ↳ Correct gradient descent
- ↳ Too slow

### • Online back-Propagation

- ↳ Stochastic gradient descent
- ↳ Fast but with fluctuations

### • Partial batched back-Propagation

- ↳ Intermediate b/w online & batched
- ↳ fast & stable
- ↳ Efficient use of hardware

# ✓ Multilayer Neural Network.

- Architecture
  - input layer
  - Output layer
  - One or more hidden layers

→ Feed forward Propagation

→ Prediction Problem

→ Training - find weight & threshold of multilayer n.v that minimize quadratic error over the training patterns

- Minimize
- Problem →  $E$  has too many local minima
  - cannot be solve by derivatives = 0

## OPTIMIZATION By

## Gradient Descent

→ Idea.

↳ Start in random position

↳ Try to move down in small step

→ formalization (minus the gradient  $-\nabla E$ )

→ Update weights & threshold to steepest descent.

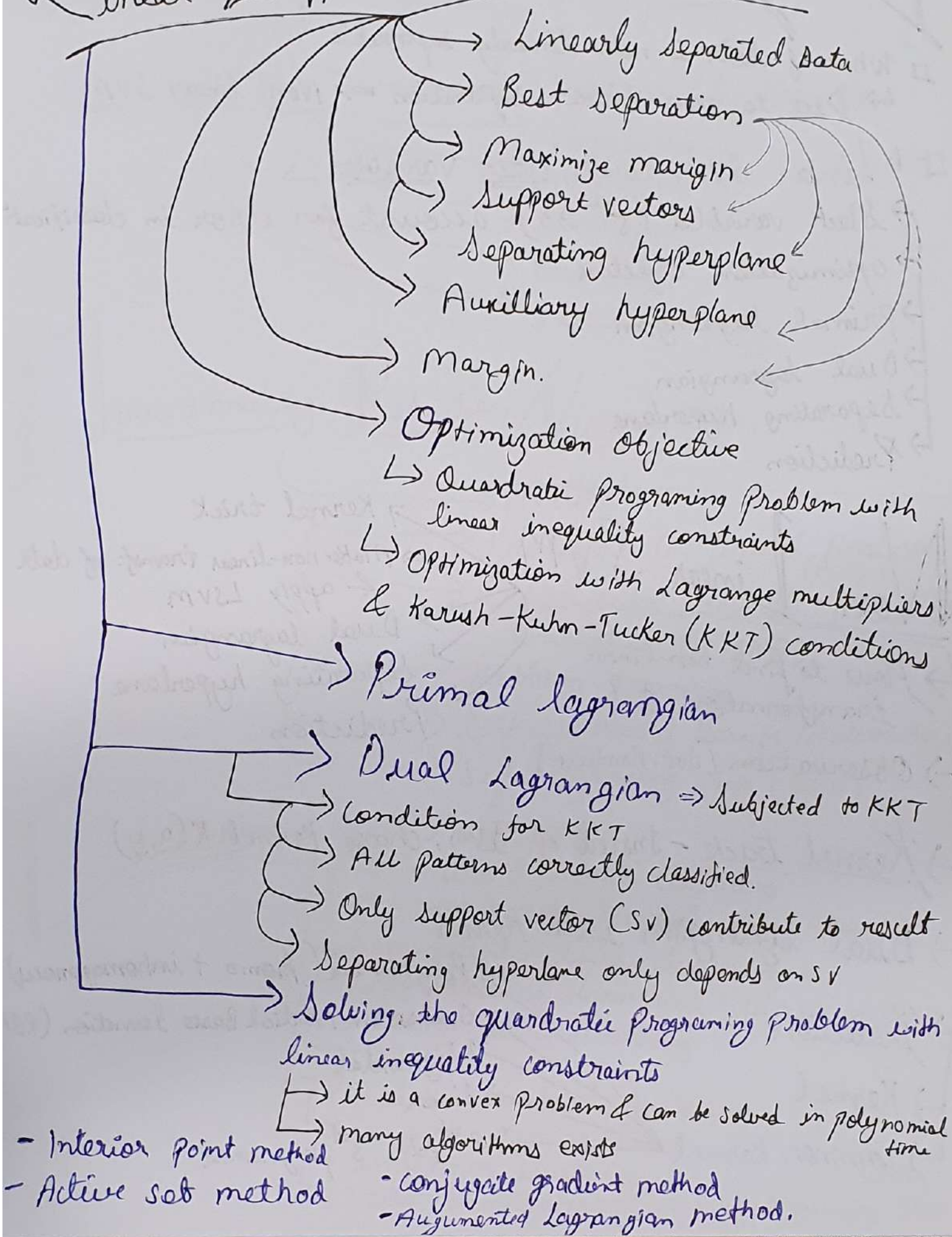
→ To ensure small step, learning rate  $\eta$  is used.

→ To avoid oscillation, include INERTIA to the movement.

- Momentum term.



# Linear Support Vector Machine (SVM)





# Linear Support Vector Machine (SVM)

□ What if data is non-linearly separable?

↳ Due to non-linear separation  $\Rightarrow$  Non-linear SVM

□  $\Rightarrow$  Linear SVM with slack variables  $\rightarrow$

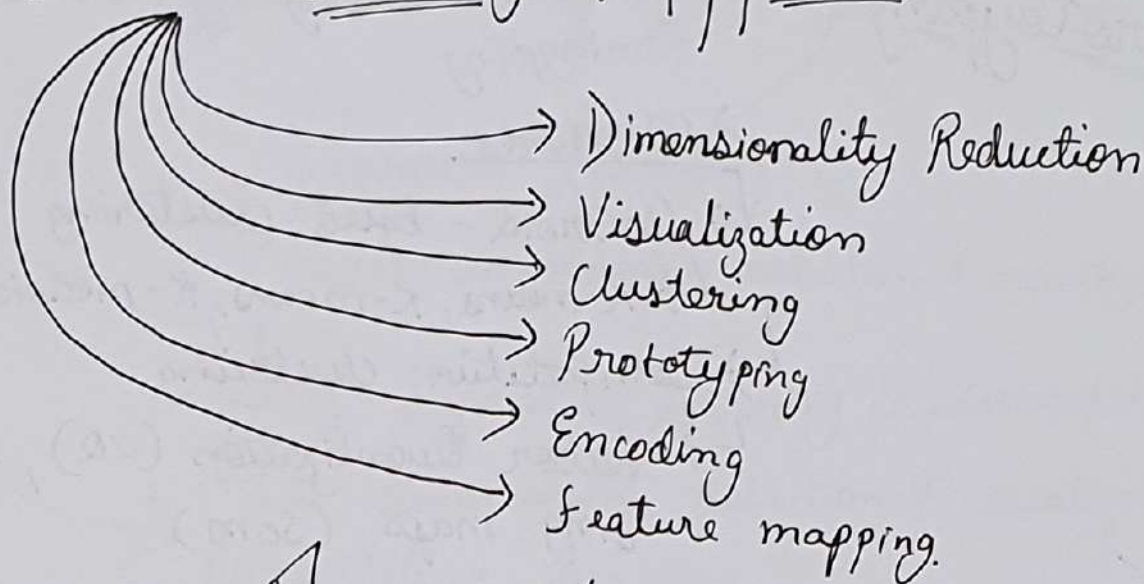
- $\rightarrow$  Slack variables  $\{F^u \geq 0\}$  account for error in classification
- $\rightarrow$  Optimization objective
- $\rightarrow$  Primal lagrangian
- $\rightarrow$  Dual lagrangian
- $\rightarrow$  Separating hyperplane
- $\rightarrow$  Prediction

## Non-Linear SVM

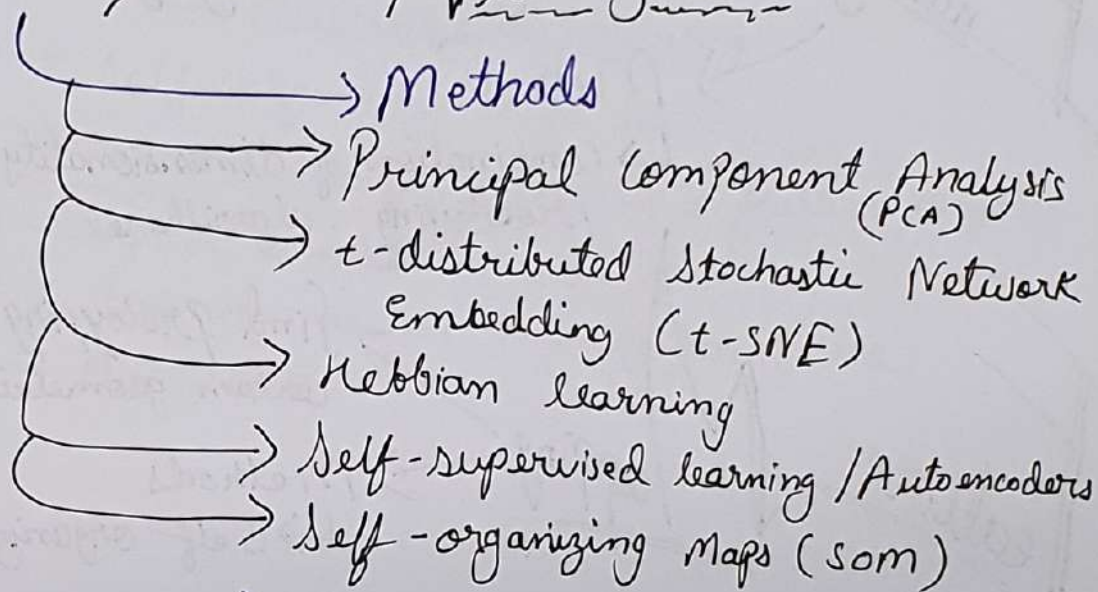
- $\rightarrow$  Kernel trick
- $\rightarrow$  Make non-linear transf. of data & apply LSVM
- $\rightarrow$  Dual lagrangian
- $\rightarrow$  Separating hyperplane
- $\rightarrow$  Prediction
- $\rightarrow$  How to find non-linear transformation  $\Phi$ ?
- $\rightarrow$  Observation [Dot Product]
- $\rightarrow$  Kernel trick - Instead of  $\Phi(x)$ , choose kernel  $K(x, y)$
- $\rightarrow$  Dual lagrangian with Kernel
- $\rightarrow$  Prediction
- $\rightarrow$  Kernel
- $\rightarrow$  Common kernel
  - $\rightarrow$  Polynomial (~~homo~~ + inhomogeneous)
  - $\rightarrow$  Gaussian / Radial Basis Function (RBF)
  - $\rightarrow$  Sigmoidal
  - $\rightarrow$  Linear
  - $\rightarrow$  2<sup>nd</sup> Poly. 3<sup>rd</sup> polynomial



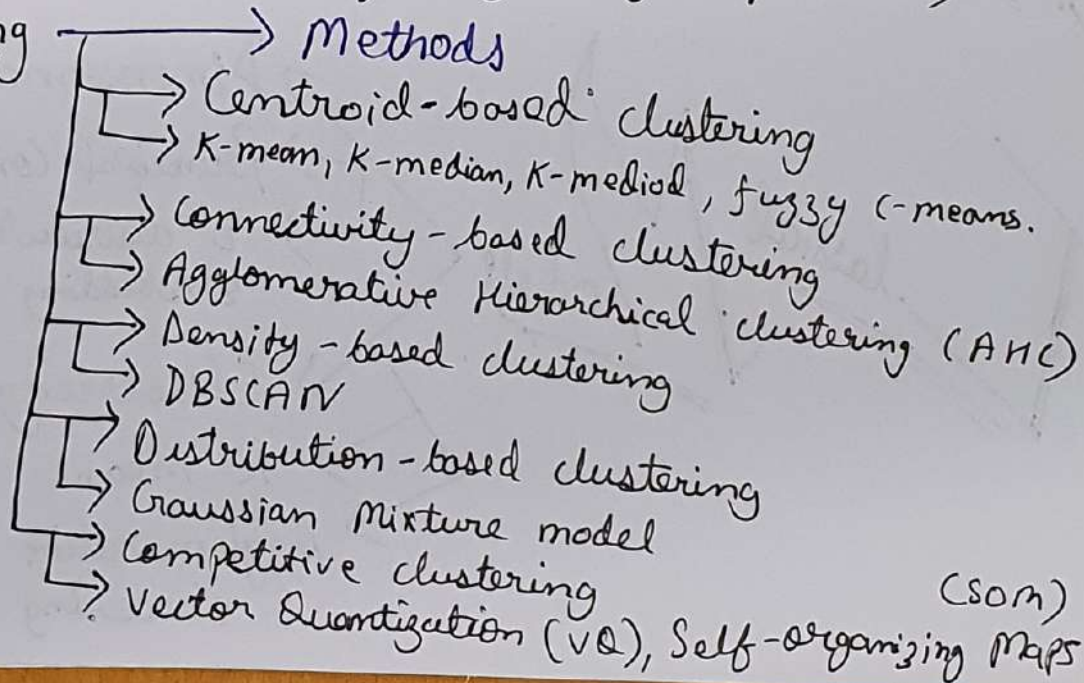
# Unsupervised Learning Approaches



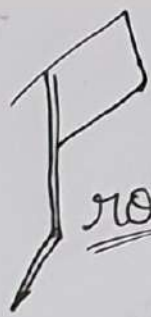
## Dimensionality Reduction / Visualization



## Clustering







## Prototyping

→ it define a clustering  $\{C_1, \dots, C_k\}$  of the original data.

→ some clustering algo. are based on Prototyping

### Methods

→ Centroid-based clustering

→ K-means, K-medoids, fuzzy C-means

→ Competitive clustering

→ Vector Quantization (VQ), Self-organizing maps (SOM)



## Encoding

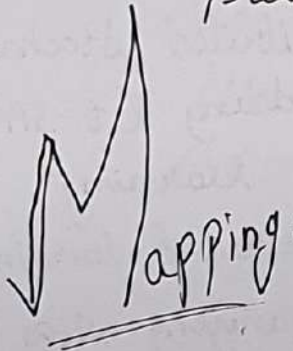
→ find lower-dimensional Prototypes & the encoding & decoding function (with loss function)

### Methods

→ Combination of dimensionality reduction & Prototyping algorithms



## Feature

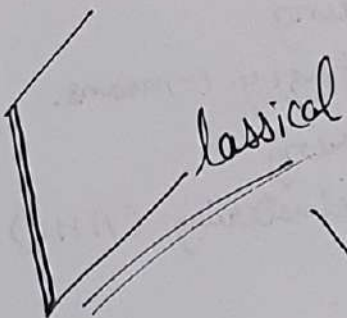


## Mapping

→ find Prototyping that preserve a certain geometric arrangement

### Methods

→ Self-organizing-Maps (SOM)



## Classical

## Models

### Dimensionality Reduction

→ Principal Component Analysis (PCA)

→ t-distributed Stochastic Network Embedding (t-SNE)

### Clustering

→ K-means

→ Agglomerative Hierarchical Clustering (AHC)



# Unsupervised Learning with Neural Network

