1. FIND-S Algorithm

Algorithm:

- 1. Initialize hypothesis `h` as the most specific.
- 2. For each positive training example:
 - Compare attributes with `h`
 - If an attribute in `h` differs, generalize it (set to `?`)
- 3. Ignore negative examples.

Advantages:

- Simple and easy to implement.
- Efficient for small datasets with only positive examples.

Disadvantages:

- Ignores negative examples.
- Assumes noise-free and complete data.

Applications:

- Concept learning in early AI systems.
- Filtering emails (only identifying spam based on known spam examples).

2. Candidate Elimination Algorithm

Algorithm:

- 1. Initialize version space: `S` = most specific, `G` = most general.
- 2. For each training example:
 - If positive: remove inconsistent hypotheses from G, generalize S.
 - If negative: remove inconsistent hypotheses from S, specialize G.
- 3. Continue until all examples are processed.

Advantages:

- Uses both positive and negative examples.

 Gives a full versior 	n space of consi	stent hypotheses.
--	------------------	-------------------

Disadvantages:

- Computationally expensive for large hypothesis spaces.
- Sensitive to noisy data.

Applications:

- Learning rules for medical diagnosis.
- Quality control (acceptable vs defective products).

3. Decision Tree (ID3 Algorithm)

Algorithm:

- 1. Select the best attribute using Information Gain.
- 2. Make that attribute the root.
- 3. Split dataset by attribute values.
- 4. Recursively apply on sub-datasets.

Advantages:

- Easy to understand and visualize.
- Works well for categorical data.

Disadvantages:

- Can overfit on noisy data.
- Biased toward attributes with many values.

Applications:

- Credit risk prediction.
- Customer churn prediction.
- Disease diagnosis.

4. Artificial Neural Network (Backpropagation)

Algorithm:

- 1. Initialize weights randomly.
- 2. Forward pass: compute output.
- 3. Backward pass: calculate error and update weights.
- 4. Repeat until convergence.

Advantages:

- Can model complex nonlinear relationships.
- Learns automatically from data.

Disadvantages:

- Requires a lot of data and computation.
- Difficult to interpret (black-box).

Applications:

- Image and speech recognition.
- Stock market prediction.
- Self-driving cars.

5. Naive Bayes Classifier

Algorithm:

- 1. Calculate prior probability of each class.
- 2. Compute likelihood of features given each class.
- 3. Use Bayes' theorem to compute posterior.
- 4. Classify based on the highest posterior.

Advantages:

- Fast and simple.
- Works well with high-dimensional data.

Disadvantages:

Applications:

- Spam filtering.

- Sentiment analysis.

- Assumes feature independence.

- Not great with correlated features.

- Text classification.
6. Document Classification using Naive Bayes
Algorithm:
Same as Naive Bayes Classifier, applied to documents.
Advantages:
- Handles large vocabulary well.
- Good accuracy for text classification tasks.
Disadvantages:
- Assumes word independence.
- Struggles with context or sarcasm.
Applications:
- News categorization.
- Email filtering.
- Classifying social media content.
7. Bayesian Network for Heart Disease Diagnosis
Algorithm:
1. Construct network using medical variables (e.g., cholesterol, BP).
2. Define dependencies and conditional probabilities.

3. Use inference to predict disease probability.

Advantages:

- Represents uncertainty and causality.
- Good for reasoning under incomplete data.

Disadvantages:

- Requires domain knowledge to build.
- Complex for large datasets.

Applications:

- Medical diagnosis systems.
- Fault detection in engineering.
- Risk assessment in finance.

8. Clustering using EM and K-Means

K-Means Algorithm:

- 1. Choose k clusters.
- 2. Assign points to nearest centroid.
- 3. Update centroids and repeat.

EM Algorithm:

- 1. Start with initial parameters.
- 2. E-step: Estimate probabilities.
- 3. M-step: Maximize likelihood by updating parameters.

Advantages:

- K-Means: Fast and easy.
- EM: Handles soft clustering and probabilities.

Disadvantages:

- K-Means: Sensitive to initialization.

Applications:

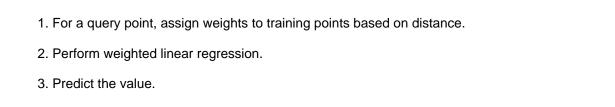
- Customer segmentation.

- Gene expression clustering.

- Image compression.

- EM: Slower and can get stuck in local minima.

9. K-Nearest Neighbors (KNN)	
Algorithm:	
1. Choose k value.	
2. Calculate distance to all training points.	
3. Pick k nearest neighbors.	
4. Assign the most common class.	
Advantages:	
- No training phase.	
- Simple and effective.	
Disadvantages:	
- Slow for large datasets.	
- Sensitive to irrelevant features and scaling.	
Applications:	
- Handwriting recognition.	
- Recommender systems.	
- Medical diagnosis.	
10. Locally Weighted Regression (LWR)	
Algorithm:	



Advantages:

- Captures local trends well.
- No need for a global model.

Disadvantages:

- Computationally expensive.
- Doesn't generalize well.

Applications:

- Robot motion planning.
- Predicting housing prices in local areas.
- Time series forecasting with local patterns.