Detailed Explanation of Algorithms Used in AI-Powered Health Connect Kiosk

The AI-Powered Health Connect Kiosk relies on a combination of machine learning (ML), deep learning, natural language processing (NLP), and optical character recognition (OCR) to deliver automated medical diagnosis and telemedicine services.

1. Naïve Bayes Algorithm (For Disease Prediction in Chatbot)

Principle:

Naïve Bayes is a probabilistic classification algorithm based on Bayes' Theorem, which states:

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

Where:

- P(A|B)P(A | B) = Probability of disease **A** given symptoms **B** (posterior probability).
- P(B|A)P(B | A) = Probability of observing symptoms **B** given disease **A** (likelihood).
- P(A)P(A) = Prior probability of disease **A** occurring.
- P(B)P(B) = Probability of observing symptoms **B** across all diseases.

Key Assumption: Symptoms are independent of each other (hence, "Naïve").

How It Works:

1. The algorithm is trained on a dataset where symptoms are mapped to diseases.
2. When a user enters symptoms into the chatbot, the system calculates posterior probabilities for all possible diseases.
3. The disease with the highest probability is selected as the predicted diagnosis.

Example Calculation:

If a patient reports symptoms: Fever, Cough.

Disease	Fever (%)	Cough (%)	Prior Probability
Flu (D1)	80%	70%	40%
COVID-19 (D2)	90%	80%	60%

Using Bayes' Theorem, we compute:

$$P(D1|Fever, Cough) = \frac{P(Fever|D1) \times P(Cough|D1) \times P(D1)}{P(Fever, Cough)}$$

$$P(D2|Fever, Cough) = \frac{P(Fever|D2) \times P(Cough|D2) \times P(D2)}{P(Fever, Cough)}$$

If P(D2) > P(D1), the system predicts COVID-19 as the most likely disease.

2. Support Vector Machine (SVM) (For Symptom Classification)

Principle:

SVM is a supervised learning algorithm that finds the best decision boundary (hyperplane) to separate different classes (diseases) in a multi-dimensional space.

A hyperplane in 2D is a line, in 3D it's a plane, and in higher dimensions, it's a mathematical function.

$$f(x) = w \cdot x + b$$

Where:

- w = Weight vector (defines the direction of separation).
- x = Input feature vector (symptoms).
- b = Bias term.

How It Works:

1. Each disease category is plotted as a point in an n-dimensional space (where n = number of symptoms).
2. The algorithm finds the hyperplane that best separates diseases.
3. When a new patient enters symptoms, SVM determines on which side of the hyperplane the input falls, predicting the corresponding disease.

3. Random Forest Algorithm (For Advanced Disease Classification)

Principle:

Random Forest is an ensemble learning method that builds multiple decision trees and combines their outputs for better accuracy.

A single Decision Tree follows:

$$\text{IF Fever} > 100^\circ\text{F AND Cough} = \text{Yes THEN COVID-19}$$

A Random Forest creates multiple trees with random subsets of data and averages their predictions for more reliable results.

How It Works:

1. Multiple decision trees are built using different subsets of patient symptom data.
2. Each tree makes an individual prediction.
3. The final diagnosis is determined by majority voting among all trees.

4. Neural Networks (MLP Classifier) (For Chatbot Decision Making)

Principle:

A Multi-Layer Perceptron (MLP) neural network consists of layers of artificial neurons that learn to recognize patterns.

A neuron processes input using:

$$y = f(w_1 x_1 + w_2 x_2 + ... + w_n x_n + b)$$

Where f is an activation function (ReLU, Sigmoid, etc.).

How It Works:

1. Input Layer: Receives symptoms (e.g., Fever, Cough, Headache).
2. Hidden Layers: Extract patterns from data.
3. Output Layer: Predicts disease type.

5. Natural Language Processing (NLP) (For Chatbot Interaction)

Principle:

NLP allows computers to understand human language using techniques like Tokenization, Named Entity Recognition (NER), and TF-IDF.

$$TF - IDF = \left(\frac{\text{Term Frequency}}{\text{Document Frequency}} \right)$$

How It Works:

1. Tokenization: Splits user input into words ("I have fever" → ["I", "have", "fever"]).
2. NER (Named Entity Recognition): Identifies medical terms.
3. TF-IDF (Term Frequency-Inverse Document Frequency): Identifies important symptoms.
4. Chatbot Response: Matches symptoms with diseases based on ML models.

6. Optical Character Recognition (OCR) – Tesseract OCR (For Medical Report Analysis)

Principle:

OCR converts scanned images of text (medical reports) into machine-readable text.

$$P(text|image) = \text{Maximum likelihood of words in the image}$$

How It Works:

1. Pre-processing: Enhances image clarity (removes noise, corrects skew).
2. Character Segmentation: Identifies individual letters/words.
3. Text Recognition: Matches characters with trained ML models to extract readable text.
4. Output Formatting: Converts extracted text into structured patient records.

Final Summary of Algorithms Used

Algorithm	Purpose	Application
Naïve Bayes	Probabilistic classification	Disease prediction
SVM	Finds optimal hyperplane	Classifies symptoms into diseases
Random Forest	Ensemble learning	Improves disease diagnosis accuracy
Neural Networks (MLP)	Deep learning	Chatbot decision-making
NLP (TF-IDF, WordNet, Tokenization)	Text processing	Symptom analysis in chatbot
OCR (Tesseract OCR)	Image-to-text conversion	Extracts medical report data