Ο κύκλος ζωής δεδομένων περιλαμβάνει τα βασικά στάδια:

1. **Συλλογή δεδομένων**
2. **Προεπεξεργασία**
3. **Ανάλυση/Εκπαίδευση μοντέλου**
4. **Αποθήκευση αποτελεσμάτων**
5. **Οπτικοποίηση**
6. **Αυτοματισμός ροής μέσω Node-RED**

node-red-dashboard (requirement)

Kane deploy

# 🔧 Τι κάνει κάθε Dockerfile;

**1. ✅ docker/Dockerfile → Python μοντέλο**

Χρησιμοποιείται για:

* Εκπαίδευση ML μοντέλου (με π.χ. scikit-learn)
* Τρέχει smoking-model.py με δεδομένα

📦 **Βασίζεται σε:** python:3.10-slim  
📁 Δομή:

bash

CopyEdit

docker/

├── Dockerfile

├── smoking-model.py

├── smoking.csv

└── requirements.txt

**2. ✅ node-server/Dockerfile → Node.js backend**

Χρησιμοποιείται για:

* Express server
* Κάνει request στον Python container
* Ενεργοποιείται από Node-RED μέσω /run-model

📦 **Βασίζεται σε:** node:18

**✅ Συμπέρασμα:**

**➕ Ναι, χρειάζεσαι 2 Dockerfile όταν:**

* Έχεις **δύο ανεξάρτητα services με διαφορετικό runtime** (π.χ. Node.js & Python)
* Θέλεις να κάνεις **διαχωρισμό ευθύνης** (π.χ. REST API vs μοντέλο)

**Βήματα για να το διορθώσεις:**

**🔹 1. Κατέβασε και εγκατέστησε το Node.js**

* Πήγαινε στη σελίδα: [https://nodejs.org/](https://nodejs.org/" \t "_new)
* **Κατέβασε** την LTS έκδοση (συνιστάται για περισσότερη σταθερότητα).
* **Εγκατέστησέ την** (θα εγκαταστήσει και το npm αυτόματα).

**🔹 2. Έλεγξε ότι όλα εγκαταστάθηκαν σωστά**

Άνοιξε **νέο PowerShell ή Command Prompt** και γράψε:

````````````````````````node -v

npm -v

Αν σου εμφανίσει εκδόσεις (π.χ. v20.11.1), είσαι έτοιμος.

**🔹 3. Εγκατάσταση Node-RED**

Τώρα μπορείς να το εγκαταστήσεις:

npm install -g --unsafe-perm node-red

**🔹 4. Εκκίνηση Node-RED**

node-red

Αν όλα πάνε καλά, θα σου εμφανίσει μήνυμα ότι τρέχει σε διεύθυνση όπως:

http://127.0.0.1:1880

**Build:**

docker build -t smoking-analyzer .

**Run:**

docker run --rm smoking-analyzer

Ara ta bhmata einai egkatastash docker kai node.js, node red.

Meta build kai run tou dockerfile

# Αναφορα

### **1. Εισαγωγή**

Η παρούσα εργασία αποσκοπεί στην ανάπτυξη ενός πλήρως αυτοματοποιημένου συστήματος πρόβλεψης καπνίσματος, βασισμένου σε μοντέλο μηχανικής μάθησης. Η υλοποίηση γίνεται με αξιοποίηση τεχνολογιών όπως Docker, Python (Flask, Scikit-learn), Node.js και Node-RED, με στόχο τη δημιουργία ενός σύγχρονου και ευέλικτου περιβάλλοντος ανάπτυξης, δοκιμής και παρουσίασης αποτελεσμάτων. !!!

### **2. Περιγραφή Συστήματος**

Το σύστημα αποτελείται από:

1. **Python Container**  
   Υλοποιεί το μοντέλο μηχανικής μάθησης, βασισμένο σε Random Forest. Εκτελείται σε Flask API με endpoint /run-model, το οποίο επιστρέφει την ακρίβεια και το classification report.
2. **Node.js API**  
   Δρα ως ενδιάμεσος server, εάν απαιτείται περαιτέρω routing ή μετασχηματισμός δεδομένων. !!!
3. **Node-RED UI**  
   Παρέχει οπτικοποίηση των αποτελεσμάτων μέσω dashboard, εμφανίζοντας την ακρίβεια και τον πίνακα με τα metrics (precision, recall, f1-score, support).

### **3. Τεχνολογίες**

| **Τεχνολογία** | **Περιγραφή** |
| --- | --- |
| **Docker** | Container και απομόνωση των εφαρμογών !!! |
| **Python** | Ανάπτυξη του μοντέλου και API μέσω Flask |
| **Scikit-learn** | Εκπαίδευση και αξιολόγηση του μοντέλου |
| **Pandas** | Διαχείριση και προεπεξεργασία δεδομένων |
| **Node-RED** | Οπτικοποίηση δεδομένων σε dashboard |
| **Node.js** | Ενδιάμεσο API server |

### **4. Περιγραφή Node-RED flow**

Η ροή αποτελείται από τα εξής βασικά βήματα:

* inject node: ενεργοποιεί την εκτέλεση
* http request node: αποστέλλει GET request στο container
* json node: μετατρέπει το response σε αντικείμενο JSON
* function node: εξάγει την ακρίβεια και μορφοποιεί το classification report
* ui\_gauge: εμφανίζει την ακρίβεια σε ποσοστό
* ui\_template: αποδίδει το report σε μορφή πίνακα

### **6. Δεδομένα & Μοντέλο**

#### Dataset:

Χρησιμοποιείται το smoking.csv, το οποίο περιλαμβάνει χαρακτηριστικά όπως:

* Ηλικία
* Φύλο
* Δείκτης Μάζας Σώματος
* Επίπεδο οξυγόνου στο αίμα (Spo2)
* Σωματική δραστηριότητα

#### Μοντέλο:

Random Forest Classifier

* Εκπαιδεύεται με μεθόδους train/test split
* Επιστρέφει ακρίβεια (accuracy) και αναλυτικό classification report (μέσω sklearn.metrics.classification\_report)

### **7. Αποτελέσματα**

A screenshot of a graph

AI-generated content may be incorrect.

### **8. Οδηγίες Εκτέλεσης**

1. Εκκίνηση containers:

docker-compose up --build

1. Άνοιγμα Node-RED: <http://localhost:1880>
2. Εισαγωγή της ροής (flow) μέσω "Import → Clipboard" !!!
3. Deploy !!!
4. Πάτημα “Run Model” για εκτέλεση
5. Εμφάνιση dashboard: <http://localhost:1880/ui>

### **9. Συμπεράσματα**

Η παρούσα υλοποίηση αποδεικνύει πως η ενοποίηση εργαλείων μηχανικής μάθησης με σύγχρονες μεθοδολογίες containerization (Docker) και low-code interfaces (Node-RED) μπορεί να προσφέρει εύχρηστα και δυναμικά εργαλεία ανάλυσης δεδομένων.

Το σύστημα είναι επεκτάσιμο και μπορεί να υποστηρίξει:

* Επιπλέον μοντέλα
* Αποθήκευση αποτελεσμάτων (π.χ. σε βάση)
* Εξαγωγή σε PDF, CSV ή email alerts

### **10. Προτάσεις για Μελλοντική Επέκταση**

* Προσθήκη δυνατότητας ανεβάσματος αρχείων CSV μέσω dashboard
* Αυτόματη αποθήκευση αναφορών
* Διασύνδεση με βάσεις δεδομένων (π.χ. PostgreSQL)
* Real-time stream επεξεργασία