

**ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΩΣ**

**ΤΜΗΜΑ ΨΗΦΙΑΚΩΝ ΣΥΣΤΗΜΑΤΩΝ**

**ΜΕΤΑΠΤΥΧΙΑΚΟ ΠΡΟΓΡΑΜΜΑ ΣΠΟΥΔΩΝ**

**«ΠΛΗΡΟΦΟΡΙΑΚΑ ΣΥΣΤΗΜΑΤΑ & ΥΠΗΡΕΣΙΕΣ»**

**ΚΑΤΕΥΘΥΝΣΗ**

**« ΠΡΟΗΓΜΕΝΑ ΠΛΗΡΟΦΟΡΙΑΚΑ ΣΥΣΤΗΜΑΤΑ»**

**ΑΠΑΛΛΑΚΤΙΚΗ ΕΡΓΑΣΙΑ ΕΞΑΜΗΝΟΥ ΓΙΑ ΤΟ ΜΑΘΗΜΑ**

**«ΕΥΦΥΗ ΠΛΗΡΟΦΟΡΙΑΚΑ ΣΥΣΤΗΜΑΤΑ ΚΑΙ ΤΕΧΝΗΤΗ ΝΟΗΜΟΣΥΝΗ»**

**ΚΑΛΟΓΕΡΟΠΟΥΛΟΣ ΓΕΩΡΓΙΟΣ, ΜΕ2453**

**ΚΑΛΟΜΟΙΡΗ ΕΡΜΙΟΝΗ, ΜΕ2455**

**ΛΑΜΠΡΟΠΟΥΛΟΥ ΜΑΡΙΑ, me2463**

**ΑΘΗΝΑ, ΙΟΥΝΙΟΣ 2025**

περιεχομενα

ΓΕΝΙΚΗ ΠΕΡΙΓΡΑΦΗ3

ΥΛΟΠΟΙΗΣΗ4

ΑΡΧΙΤΕΚΤΟΝΙΚΗ5

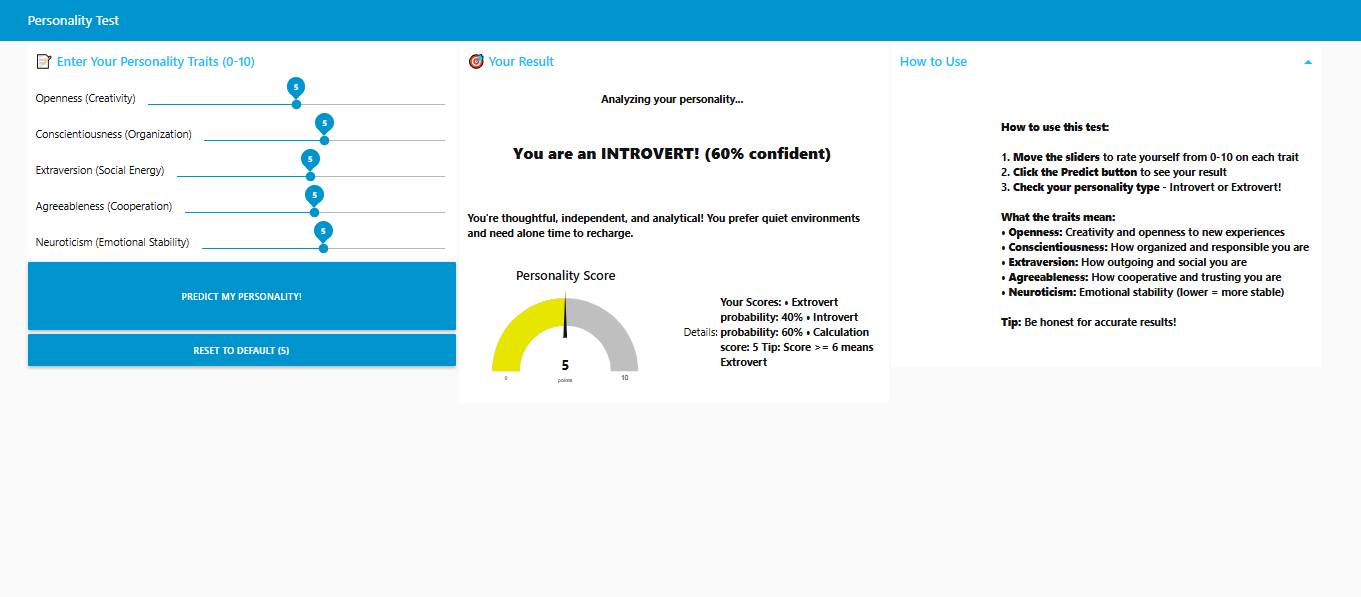
ΤΕΧΝΟΛΟΓΙΕΣ6

ΓΕΝΙΚΗ ΠΕΡΙΓΡΑΦΗ

Η παρούσα εργασία εκπονήθηκε στο πλαίσιο του μαθήματος «Ευφυή Πληροφοριακά Συστήματα και Τεχνητή Νοημοσύνη». Το θέμα που επιλέξαμε σαν ομάδα ήταν το θέμα **«10. Analytics as a Service».**

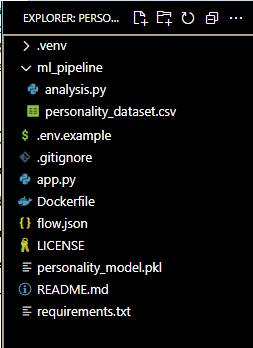
Το dataset που επιλέχθηκε μέσω του kaggle.com ([link](https://www.kaggle.com/datasets/rakeshkapilavai/extrovert-vs-introvert-behavior-data)) με ονομασία “Extrovert vs. Introvert Behavior Data”. Περιέχει δεδομένα χρηστών και χαρακτηριστικά προσωπικότητας. Αποτελεί ιδανικό σύνολο για την εφαρμογή predictive analytics και την εξαγωγή χρήσιμων συμπερασμάτων σχετικά με το προφίλ κάθε χρήστη.

Ως ομάδα, υλοποιήσαμε μία end-to-end πλατφόρμα παροχής αναλύσεων βασισμένων σε μηχανική μάθηση, αξιοποιώντας τεχνικές data science, RESTful APIs και οπτικοποίηση μέσω Node-RED. Η ομάδα υιοθέτησε συνεργατική προσέγγιση επιτυγχάνοντας μια σαφή κατανομή ευθυνών και καθαρές ροές εργασίας.

Η τελική υλοποίηση συνδυάζει μηχανική μάθηση, API τεχνολογίες και low-code οπτική πλατφόρμα σε ένα ενιαίο, modular και επεκτάσιμο σύστημα ανάλυσης δεδομένων. Η εργασία αναδεικνύει τις δυνατότητες συνεργασίας ετερογενών τεχνολογιών για την παροχή Analytics as a Service, με πρακτική εφαρμογή σε πραγματικά datasets. Έτσι δημιουργήθηκε ένα test προσωπικότητας το οποίο μπορεί να δείξει στον χρήστη βάση το πόσο ισχυρό ή όχι είναι ένα χαρακτηριστικό αν είναι εξωστρεφής ή εσωστρεφής.

NODE RED UI ΑΡΧΙΚΗ

# υλοποιηση

Αρχικά έγινε προκαταρκτική ανάλυση δεδομένων, επιλογή και επεξεργασία χαρακτηριστικών, εκπαίδευση μοντέλων και αξιολόγησή τους. Χρησιμοποιήθηκε η βιβλιοθήκη Scikit– learn για την ανάπτυξη του μοντέλου, το οποίο στη συνέχεια εξήχθη σε μορφή συμβατή με παραγωγικά συστήματα.

Επόμενο βήμα, ήταν να αναπτύξουμε RESTful API με χρήση Flask, το οποίο επιτρέπει την επικοινωνία του backend με το εξωτερικό σύστημα. Η υπηρεσία περιλαμβάνει endpoints για αποστολή δεδομένων, πρόβλεψη αποτελεσμάτων και διαχείριση αιτημάτων. Επίσης, υλοποιήθηκε containerization μέσω Docker, διασφαλίζοντας φορητότητα και συνέπεια στο περιβάλλον εκτέλεσης.

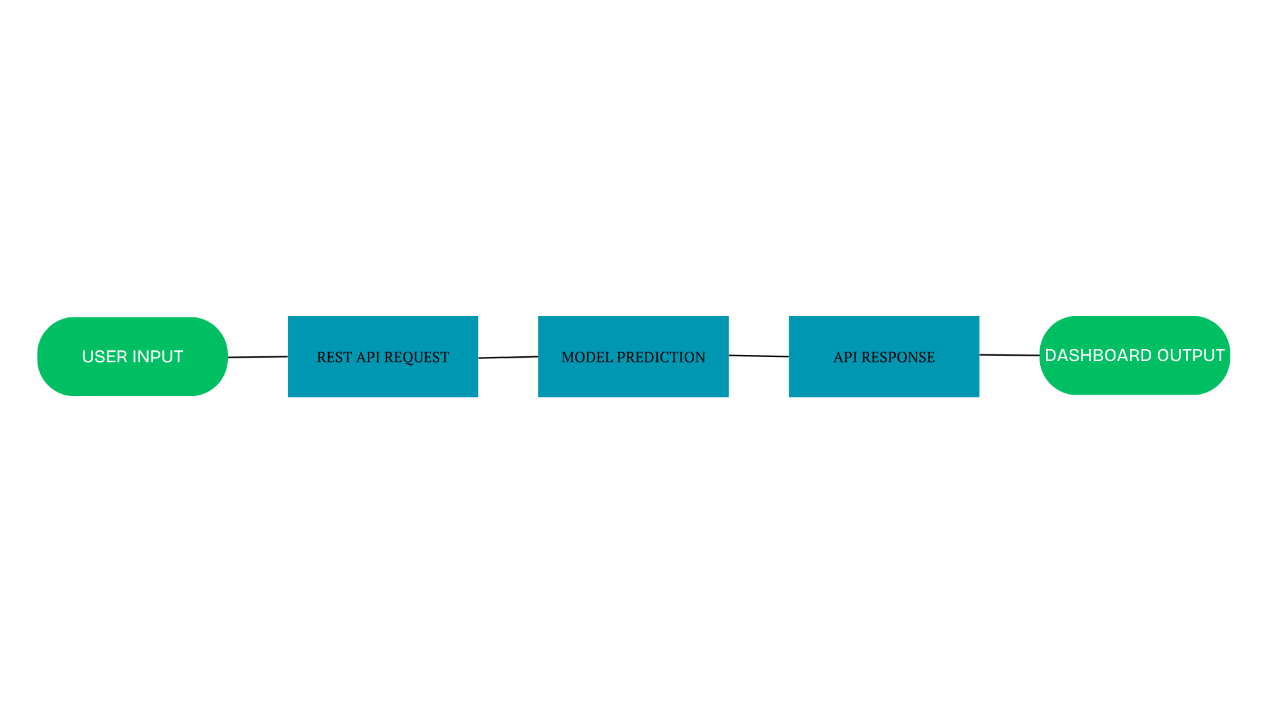
ΔΟΜΗ ΠΡΟΓΡΑΜΜΑΤΟΣ ΣΤΟ VS CODE

Τέλος, χρησιμοποιήσαμε την πλατφόρμα Node-RED [(link)](http://localhost:1880) για την υλοποίηση της ροής εργασιών και της διεπαφής χρήστη. Η λύση περιλαμβάνει dashboard για οπτικοποίηση των προβλέψεων, καθώς και κόμβους παρακολούθησης και debugging. Από το UI ([link](http://localhost:1880/ui)), ο τελικός χρήστης μπορεί να εισάγει δεδομένα επιλέγοντας χαρακτηριστικά, να λαμβάνει αποτελέσματα (αν είναι εσωστρεφής ή εξωστρεφής) και να παρακολουθεί τη λειτουργία της υπηρεσίας σε πραγματικό χρόνο πατώντας τα αντίστοιχα κουμπιά.

# ΑΡΧΙΤΕΚΤΟΝΙΚΗ

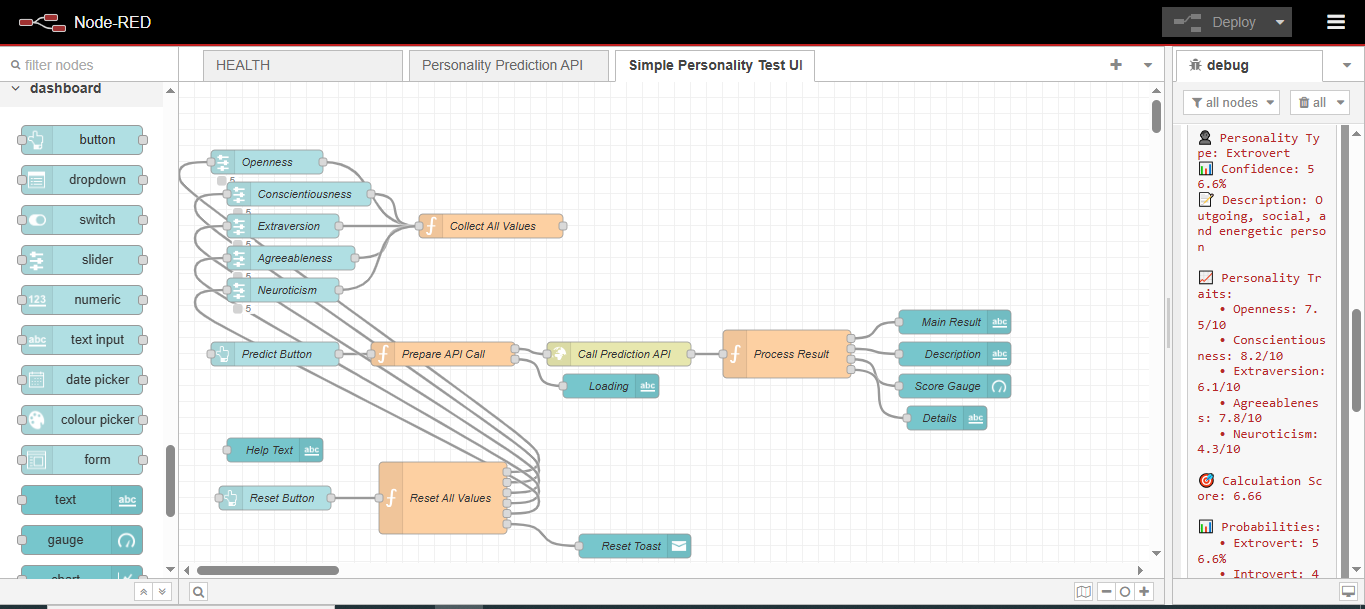
Όσον αφορά την αρχιτεκτονική, το έργο ακολουθεί modular προσέγγιση με τρία βασικά υποσυστήματα: μηχανική μάθηση (ML pipeline), διασύνδεση μέσω RESTful API, και οπτικοποίηση & ροές μέσω Node-RED. Κάθε υποσύστημα αναπτύχθηκε σε ξεχωριστό φάκελο (ml\_pipeline/, api/, workflows/) και ενσωματώθηκε σε ένα κοινό workflow στο github.

Η ροή ξεκινάει με την επεξεργασία των δεδομένων και την εκπαίδευση του μοντέλου μέσω Scikit – learn. Το εκπαιδευμένο μοντέλο αποθηκεύεται τοπικά σε μορφή .h5 και στη συνέχεια φορτώνεται από το API backend. Το API, υλοποιημένο με Flask, παρέχει endpoints για την αποστολή δεδομένων και την επιστροφή προβλέψεων. Το όλο σύστημα συσκευάζεται με Docker, διασφαλίζοντας κοινό περιβάλλον εκτέλεσης ανεξάρτητα από την πλατφόρμα.

Η διεπαφή χρήστη έχει δημιουργηθεί με χρήση Node-RED, προσφέροντας έναν εύχρηστο τρόπο αποστολής δεδομένων στο API και λήψης αποτελεσμάτων, μέσω dashboard και flow κόμβων. Ο τελικός χρήστης μπορεί να χρησιμοποιήσει την υπηρεσία χωρίς καμία γνώση προγραμματισμού, απλώς συμπληρώνοντας τα απαιτούμενα πεδία και βλέποντας τα αποτελέσματα σε πραγματικό χρόνο.

FLOWCHART ΣΥΣΤΗΜΑΤΟΣ

# ΤΕΧΝΟΛΟΓΙΕΣ

* VS Code για ανάπτυξη του κώδικα.
* Python για την ανάπτυξη του μοντέλου και σύνδεση API μέσω Flask.
  + H βιβλιοθήκη Pandas (Διαχείριση – επεξεργασία δεδομένων).
  + H βιβλιοθήκη Scikit – learn ( Εκπαίδευση – αξιολόγηση μοντέλου).
* Docker ως container.
* Node.js - NodeRed για ενδιάμεσο server του API και την οπτικοποίηση του συστήματος.
* Github για συλλογή του κώδικα.

NODE RED DASHBOARD