



Αριστοτέλειο Πανεπιστήμιο Θεσσαλονίκης

Πολυτεχνική Σχολή

Τμήμα Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών

Τομέας Ηλεκτρονικής και Υπολογιστών

ΑΝΑΠΤΥΞΗ ΠΡΟΤΥΠΟΥ ΣΤΑΘΜΟΥ ΠΑΡΑΚΟΛΟΥΘΗΣΗΣ ΚΑΙ ΒΕΛΤΙΣΤΟΠΟΙΗΣΗΣ ΚΑΛΛΙΕΡΓΕΙΩΝ

Μαρία Ξουρή

ΑΕΜ: 10240

Υπεύθυνος καθηγητής: Παυλίδης Βασίλειος, Αναπληρωτής
Καθηγητής

Περίληψη

Η ανάπτυξη του Διαδικτύου των Πραγμάτων (Internet Of Things (IoT)) έχει παίξει καθοριστικό ρόλο στη βελτιστοποίηση πολλών τομέων της κοινωνικής πραγματικότητας. Ο αγροτικός τομέας αποτελεί έναν από αυτούς με την εισχώρηση του IoT τόσο στον τομέα της γεωργίας όσο και στην κτηνοτροφία. Παρόλα αυτά, η παρακολούθηση των παραμέτρων των γεωργικών καλλιεργειών με τη χρήση του διαδικτύου των πραγμάτων δεν είναι τόσο διαδεδομένη όσο θα μπορούσε καθώς οι καλλιεργητές προτιμούν έναν πιο «εμπειρικό» τρόπο διαχείρισης των καλλιεργειών τους. Η διπλωματική αυτή έχει στόχο να παρουσιάσει έναν τρόπο παρακολούθησης της καλλιέργειας της ντομάτας και της πιπεριάς καθώς και να δημιουργήσει ένα μοντέλο Μηχανικής Μάθησης που θα διαχωρίζει τις ασθένειες που μπορεί να αναπτύξουν τα φυτά με βάση τις μετρούμενες παραμέτρους της καλλιέργειας και ακολούθως, θα προβλέπει την υγεία των φυτών βάσει των δεδομένων του περιβάλλοντος.

Abstract

The development of Internet of Things (IoT) has played a major role in optimizing many sectors of our society. Farming is one of them with the use of IoT in both agriculture and animal husbandry. Despite the development of IoT, the crop's health monitoring is not as popular for farmers as it could be because they are used to evaluate the crop's health based on past experience. This thesis project has the primary goal to introduce a specific way of monitoring the tomato and pepper cultivation and to create a Machine Learning Model that separates the diseases the cultivation might develop based on measured characteristics of the crop. The main challenge is also to forecast the health of the plants based on the environmental data in the next days.

Ευχαριστίες

Θα ήθελα να ευχαριστήσω τα άτομα που με έχουν στηρίξει στην σχολή όπως την οικογένειά μου, τους φίλους μου και φυσικά τον υπεύθυνο καθηγητή που με βοήθησε πολύ με τις οδηγίες και την μεγάλη του πρόθεση για την δημιουργία μιας πολύπλευρης εργασίας.

Περιεχόμενα

1. Εισαγωγή.....	10
1.1 Πλεονεκτήματα της χρήσης του IoT στη βελτιστοποίηση των γεωργικών καλλιεργειών	10
1.2 Βαθμός αξιοποίησης του IoT από τους καλλιεργητές και μελλοντικές προκλήσεις.....	12
1.2.1 Πραγματοποίηση συνέντευξης στους καλλιεργητές σε σχέση με το IoT..	12
1.2.2 Προκλήσεις.....	13
1.2.3 Βαθμός της χρήσης του Ag-IoT	13
1.3 Κατηγορίες των Ag-IoT sensors	13
1.3.1 Ακουστικοί	14
1.3.2 Βιολογικοί	14
1.3.3 Χημικοί.....	14
1.3.4 Ηλεκτρικοί.....	14
1.3.5 Μηχανικοί.....	14
1.3.6 Οπτικοί.....	14
1.3.7 Θερμικοί.....	14
1.4 Σκοπός της διπλωματικής.....	15
1.5 Διάρθρωση της διπλωματικής.....	15
2. Ανάπτυξη του Πρότυπου Σταθμού Παρακολούθησης και Βελτιστοποίησης Καλλιεργειών	17
2.1 Διαδικασία συνδεσμολογίας και λήψης λογισμικού του σταθμού.....	17
2.2 Υλικό/Εξαρτήματα και τα χαρακτηριστικά τους.....	21
2.2.1 Raspberry Pi 3 Model B+.....	21
2.2.2 Grove Base Hat for Raspberry Pi	22
2.2.3 Grove – Moisture Sensor	24
2.2.4 Grove - Light Sensor	25
2.2.5 Grove - Temperature & Humidity Sensor (DHT11)	25
2.2.6 SD card.....	25
2.2.7 ESP8266MOD VENOR	26
2.3 Λογισμικό του Σταθμού Παρακολούθησης.....	27
2.3.1 Λογισμικό του Grove - Temperature & Humidity Sensor.....	27
2.3.2 Λογισμικό του Grove - Light Sensor	28
2.3.3 Λογισμικό του Grove-Moisture Sensor	29
2.4 Πλεονεκτήματα και μειονεκτήματα της χρήσης του Pi 3B+.....	30
3. Μετρούμενα Χαρακτηριστικά Καλλιεργειών	31

3.1	Ανάλυση των μετρούμενων παραμέτρων και σκοπός της μέτρησης	31
3.2	Πρόγραμμα Μηχανικής Μάθησης για την αξιολόγηση της υγείας των φυτών ...	33
3.3	Συμπεράσματα για την υγεία της πιπεριάς.....	35
3.4	Συμπεράσματα για την υγεία της ντομάτας.....	38
4.	Βελτιστοποίηση της καλλιέργειας μέσω δημιουργίας μοντέλου μηχανικής μάθησης για την πρόβλεψη πιθανής ασθένειας.....	42
4.1	Επίλογή μοντέλου μηχανικής μάθησης	43
4.2	Πλεονεκτήματα του Plant Disease Forecasting και διαδικασία υλοποίησης.....	44
4.2.1	Χρήση μοντέλου Μηχανικής Μάθησης	44
4.2.2	Αποτελέσματα	48
4.2.3	Δεδομένα καιρού	52
4.2.4	Πρόβλεψη πιθανών ασθενειών	54
5.	Αποτελέσματα και εργασίες	57
5.1	Αποτελέσματα	57
5.2	Εργασίες	57
	Κώδικας.....	58

Κατάλογος Σχημάτων

- 1.1 Διαδικασία επεξεργασίας δεδομένων από το Data Analysis [1]
- 1.2 Απόψεις των καλλιεργητών για το IoT [2]
- 2.1 Δημιουργία σταθμού καλλιεργειών [3]
- 2.2 Προσθήκη ονόματος χρήστη και κωδικού στο λειτουργικό του Pi [4]
- 2.3 Προσθήκη της εντολής για την είσοδο στο Pi [5]
- 2.4 Χρήση της VNC για την σύνδεση στο Desktop [6]
- 2.5 Διαδικασία δημιουργίας του σταθμού καλλιεργειών [7]
- 2.6 Προσθήκη του Pi στα φυτά για τις μετρήσεις [8]
- 2.7 Στοιχεία του Pi [9]
- 2.8 Σύνδεση του Pi με ESP και με Ethernet [10]
- 2.9 Λογισμικό του Grove - Temperature & Humidity Sensor [11]
- 2.10 `seed_dht` [12]
- 2.11 Λογισμικό του Grove - Light Sensor [13]
- 2.12 Λογισμικό του Grove-Moisture Sensor [14]
- 2.13 `grove_moisture_sensor` [15]
- 3.1 Μετρήσεις παραμέτρων και αξιολόγηση της υγείας της ντομάτας [16]
- 3.2 Αποτελέσματα των μετρήσεων [17]
- 3.3 Διαδικασία του Import [18]
- 3.4 Διαδικασία του Training [19]
- 3.5 Διαδικασία του Use [20]
- 3.6 Ασθένειες πιπεριάς: Μαύρο σημάδι και ωίδιο [21]
- 3.7 Μεταβολή της ακτινοβολίας του φωτός σε σχέση με τις ημέρες [22]
- 3.8 Μεταβολή της θερμοκρασίας σε σχέση με τις ημέρες [23]
- 3.9 Μεταβολή της υγρασίας του αέρα σε σχέση με τις ημέρες [25]
- 3.10 Μεταβολή υγρασίας του εδάφους σε σχέση με τις ημέρες [26]
- 3.11 Διαχωρισμός ασθενειών [26]
- 3.12 Ντομάτα με και χωρίς Yellow Leaf Curl [27]
- 3.13 Target Spot και Early Blight [28]
- 3.14 *Bacterial spots [29]

- 3.15 Septoria Leaf Spot, Late Blight [30]
- 3.16 Mosaic Virus και Leaf Mold [31]
- 4.1 k-means clustering [32]
- 4.2 Διαδικασία δημιουργίας του μοντέλου [33]
- 4.3 Φόρτωση δεδομένων [34]
- 4.4 Κανονικοποίηση [35]
- 4.5 Principal Component Analysis [36]
- 4.6 Silhouette Method [37]
- 4.7 Silhouette Scores για διαφορετικές τιμές των συστάδων [38]
- 4.8 Αποτελέσματα του μοντέλου [39]
- 4.9 Συστάδες 1 – 10 [40]
- 4.10 Συστάδες 1 – 5 [41]
- 4.11 Συστάδες 6, 7 [42]
- 4.12 Συστάδες 8, 9, 10 [43]
- 4.13 Μεταβλητές για την χρησιμοποίηση του url [44]
- 4.14 Τμήμα των αποτελεσμάτων των μεταβλητών page και soup [45]
- 4.15 Διαχωρισμός κειμένου [46]
- 4.16 Τμήμα των αποτελεσμάτων σε μορφή κειμένου [47]
- 4.17 Διαχωρισμός των χρήσιμων δεδομένων [48]
- 4.18 Δημιουργία πίνακα με τις παραμέτρους [49]
- 4.19 Διαχωρισμός των στοιχείων [50]
- 4.20 Μετατροπή σε float και υπολογισμός του μέσου [51]
- 4.21 Διαστήματα της υγείας του [52]
- 4.22 Παρουσίαση του αποτελέσματος στον καλλιεργητή [53]
- 4.23 Πλεονεκτήματα [54]

Κατάλογος Πινάκων

2.1 Χαρακτηριστικά του Pi 3B+ [1]

2.2 Χαρακτηριστικά του Grove Base Hat for Raspberry Pi [2]

2.3 Χαρακτηριστικά του Grove-Moisture Sensor [3]

2.4: Χαρακτηριστικά του Grove - Light Sensor [5]

2.5: Χαρακτηριστικά του Temperature & Humidity Sensor [6]

2.6: Χαρακτηριστικά του ESP8266MOD VENOR [7]

Κεφάλαιο 1

Εισαγωγή

Το Διαδίκτυο των Πραγμάτων έχει εισχωρήσει σε μεγάλο βαθμό στην ανθρώπινη ζωή καθιστώντας όλο και πιο επιτακτική την ανάγκη για την κατάκτηση του γνωστικού υποβάθρου που το συνοδεύει. Ο τομέας της γεωργίας έχει καταφέρει να χρησιμοποιήσει το Internet Of Things σε αρκετά μεγάλο βαθμό. Η ευρεία του χρήση έχει οδηγήσει στην βελτιστοποίηση των καλλιεργειών λόγω της αύξησης της παραγωγής και της αποφυγής των ασθενειών των φυτών [1].

Το IoT περιλαμβάνει εργαλεία όπως τους αισθητήρες (sensors), οι οποίοι χρησιμοποιούνται ευρέως στον γεωργικό τομέα. Βασικός τους στόχος είναι να ανιχνεύουν διάφορες φυσικές παραμέτρους και να τις μετατρέπουν σε άλλους, όπως σήματα ή αριθμούς με τους οποίους ο καλλιεργητής θα μπορεί να βγάλει συμπεράσματα για την κατάσταση των φυτών του.

1.1 Πλεονεκτήματα της χρήσης του IoT στη βελτιστοποίηση των γεωργικών καλλιεργειών

Η χρήση του IoT στον τομέα των καλλιεργειών (Ag-IoT) περιλαμβάνει πολλά πλεονεκτήματα όσον αφορά την παραγωγή των σοδειών και την ευημερία των παραγωγών.

Ένα βασικό πλεονέκτημα της χρήσης του IoT είναι η απόκτηση και η αποθήκευση των δεδομένων μεγάλης παραγωγής χωρίς την παρουσία των καλλιεργητών. Μια μεγάλη καλλιέργεια απαιτεί πολλούς IoT sensors οι οποίοι μετρούν φυσικές παραμέτρους και τις μετατρέπουν σε αριθμητικά ή άλλων ειδών δεδομένα. Τα δεδομένα αυτά αποθηκεύονται έτσι ώστε αργότερα να χρησιμοποιηθούν από τους καλλιεργητές για την αξιολόγηση της υγείας των φυτών τους, για την πρόβλεψη της ποσότητας που θα παραχθεί ή και για τη γνωστοποίηση των δεδομένων τους σε επαγγελματικούς συνεργάτες.

Η διαδικασία που αναφέρθηκε, γινόταν με εμπειρικό τρόπο τα προηγούμενα χρόνια. Παρόλο που πολλοί καλλιεργητές έχουν την ικανότητα να αξιολογήσουν επαρκώς την υγεία των καλλιεργειών τους με αυτόν τον τρόπο, μια σοδειά που καλύπτει αρκετά μεγάλη έκταση δεν είναι πάντα εύκολη υπόθεση να αξιολογηθεί. Παράλληλα, πολλά άτομα που είναι νέα στον χώρο δεν έχουν την ικανότητα να αξιολογήσουν με βάση την εμπειρία την κατάσταση των σοδειών τους και γι' αυτό το IoT τους παρέχει δεδομένα τα οποία θα τους οδηγήσουν σε πιο ασφαλή μονοπάτια.

Η χρήση του IoT έχει ένα πολύ βασικό πλεονέκτημα το οποίο είναι η απομακρυσμένη χρήση του. Σε περίπτωση που οι καλλιεργητές απουσιάζουν οι μετρήσεις μπορούν να πραγματοποιηθούν χωρίς την φυσική τους παρουσία. Επιπλέον, μπορούν να τους σταλούν δεδομένα από τις μετρήσεις έτσι ώστε να γνωρίζουν την κατάσταση της σοδειάς τους.

Έχει υπολογισθεί ότι ο πληθυσμός της γης μέχρι το 2050 θα αυξηθεί σημαντικά γεγονός που καθιστά δύσκολη την ικανοποίηση των αναγκών ενός τόσο μεγάλου αριθμού ατόμων. Για το λόγο αυτό το IoT θα μπορεί να βοηθήσει στην επιτέλεση γεωργικών εργασιών και στην παραγωγή αποτελεσμάτων γρηγορότερα και πιο αποτελεσματικά ικανοποιώντας έτσι τις ανάγκες όλων των ατόμων [3].

Το Internet of Things στον τομέα της καλλιέργειας της γης μπορεί να χρησιμοποιηθεί σε πολλές πτυχές. Πιο συγκεκριμένα, λόγω του μεγάλου πλήθους των φυσικών παραμέτρων που μπορούν να μετρηθούν, όπως θερμοκρασία, υγρασία, φωτεινότητα, συγκέντρωση χημικών στοιχείων κ.α. παράγεται μια μεγάλη ποσότητα δεδομένων. Το γεγονός αυτό έχει οδηγήσει στην ανάπτυξη του τομέα του Data Analysis. Το Data Analysis στον τομέα των καλλιεργειών, χρησιμοποιεί δεδομένα που διακρίνονται σε: δομημένα, όπως αριθμητικά στοιχεία και αδόμητα όπως φωτογραφίες και βίντεο [1]. Στο Σχήμα 1.1 αναφέρονται οι λειτουργίες του Data Analysis.



Σχήμα 1.1: Διαδικασία επεξεργασίας δεδομένων από το Data Analysis [1]

1.2 Βαθμός αξιοποίησης του IoT από τους καλλιεργητές και μελλοντικές προκλήσεις

Ο τομέας των καλλιεργειών είναι αξιοποιήσιμος κυρίως από ανθρώπους μεγαλύτερων ηλικιών που για πολλά χρόνια δεν είχαν την τεχνολογία με το μέρος τους. Για το λόγο, αυτό η χρήση του Internet of Things είναι περιορισμένη για τη συγκεκριμένη ηλικιακή ομάδα.

1.2.1 Πραγματοποίηση συνέντευξης στους καλλιεργητές σε σχέση με το IoT

Ύστερα από έρευνες που πραγματοποιήθηκαν στον Καναδά -μια χώρα με σπουδαία ανάπτυξη στη γεωργική παραγωγή- οι αγρότες ερωτήθηκαν για το πώς αξιολογούν οι ίδιοι το IoT και πόσο το χρησιμοποιούν [1]. Οι αγρότες διχάστηκαν στις απαντήσεις τους, καθώς υπήρχε ένα μεγάλο ποσοστό που επιδοκίμαζε το IoT χαρακτηρίζοντάς το ως «Βασικό τους βοηθό», ενώ οι υπόλοιποι συμμετέχοντες το χαρακτήρισαν ως έναν «Αναξιόπιστο Υπάλληλο». Παρακάτω, στο Σχήμα 1.2 παρουσιάζονται κωδικοποιημένα οι απαντήσεις των καλλιεργητών.

Προσέγγιση υπέρ του IoT	Προσέγγιση κατά του IoT
Μέτρηση με μεγαλύτερη ακρίβεια	Αναξιόπιστο
Ανταλλαγή πληροφοριών	Φόβος για βλάβη
Εξοικονόμηση χρόνου	Φόβος για απώλεια πληροφοριών
Λιγότερη εργασία	Ακριβό
Άμεση πρόσβαση	Προκαλεί σύγχυση λόγω άγνοιας
Απομακρυσμένη εργασία	

Σχήμα 1.2: Απόψεις των καλλιεργητών για το IoT [2]

Όπως παρουσιάζεται και στο σχήμα 1.2 υπάρχουν προσεγγίσεις του IoT που διαφέρουν σημαντικά μεταξύ τους. Η θετική προσέγγιση ορίζει το IoT ως ένα μέσο που διευκολύνει την παραγωγή των καλλιεργητών σε πολλαπλά πλαίσια. Με τη χρήση του IoT οι μετρήσεις

πραγματοποιούνται με μεγαλύτερη ακρίβεια καθώς περιέχουν κατά βάση αριθμητικά στοιχεία και όχι εμπειρικό χαρακτήρα. Παράλληλα, μπορεί να γίνει με ευκολία η μεταφορά δεδομένων από τον παραγωγό σε επαγγελματικούς συνεργάτες ή σε συναδέλφους. Η χρήση επιπλέον, συμβάλλει στη μείωση του χρόνου και της εργασίας καθώς πολλές εργασίες τις πραγματοποιούν τα τεχνολογικά εργαλεία χωρίς την παρέμβαση του παραγωγού.

Από την άλλη, ορισμένοι καλλιεργητές χαρακτηρίζουν το IoT ως αναξιόπιστο. Αυτό οφείλεται στο γεγονός ότι υπάρχει περίπτωση να χαθούν πληροφορίες που έχουν σημασία για την έκβαση της παραγωγής. Επιπλέον, η απουσία τεχνικών γνώσεων τους έχει οδηγήσει στον φόβο για επικαιροποίηση. Ο φόβος για τον εκμοντερνισμό όπως η χρήση κάποιου τεχνολογικού εργαλείου είναι η βασική αιτία της «σύγχυσης» των καλλιεργητών και της δυσπιστίας τους απέναντι στον τομέα του Διαδικτύου των Πραγμάτων. Επιπρόσθετα, η χρήση του IoT συνεπάγεται και απώλεια χρημάτων τόσο για την προμήθειά των εργαλείων όσο και για την συντήρησή τους.

1.2.2 Προκλήσεις

Οι προκλήσεις που αντιμετωπίζει το IoT σχετίζονται σε αρκετά μεγάλο βαθμό με την πρόσβαση στο διαδίκτυο. Πολλές περιοχές δεν μπορούν να κάνουν χρήση του διαδικτύου και κατά συνέπεια, δεν μπορούν να εφαρμόσουν την τεχνολογία των IoT. Παρόμοιο πρόβλημα αντιμετωπίζουν και οι περιοχές που έχουν έλλειψη ηλεκτρισμού. Πολλές καλλιέργειες μεγάλης κλίμακας βρίσκονται μακριά από αστικές περιοχές γεγονός που καθιστά δύσκολη την πρόσβαση στο διαδίκτυο και στον ηλεκτρισμό.

Παράλληλα, μια μεγάλη πρόκληση που αντιμετωπίζει το IoT είναι η απουσία προσωπικού που γνωρίζει τον χειρισμό του. Ο τομέας της Τεχνητής Νοημοσύνης και του IoT έχει έλλειψη από καταρτισμένο προσωπικό κι έτσι δεν μπορεί να εκπληρώσει τις ανάγκες της κοινωνίας. Επιπλέον, από περιβαλλοντικής άποψης η παραγωγή αρκετών εργαλείων χρειάζεται αρκετά υπολογίσιμη ποσότητα άνθρακα γεγονός που επιβαρύνει την περιβαλλοντική ευημερία [3].

1.2.3 Βαθμός της χρήσης του Ag-IoT

Ο βαθμός της χρήσης του Ag-IoT διαφέρει από περιοχή σε περιοχή λόγω των αναγκών που παρουσιάζει η καθεμία. Περιοχές που στηρίζουν την οικονομία τους κατά μεγάλο βαθμό στην καλλιέργεια της γης όπως η Ασία έχουν μεγαλύτερες απαιτήσεις σε IoT απ' ό,τι άλλες. Η περιοχή με την ελάχιστη έρευνα είναι η Αυστραλία. Στο σύνολο, πάνω από 20 χώρες του κόσμου χρησιμοποιούν σε ικανοποιητικό βαθμό το Ag-IoT [1].

1.3 Κατηγορίες των Ag-IoT sensors

Οι Ag-IoT sensors είναι υπεύθυνοι στο να μετατρέπουν σήματα που προέρχονται από τη φύση, σε μορφή που είναι ικανή να την επεξεργαστεί ο καλλιεργητής. Η επιλογή του κατάλληλου Ag-IoT sensor γίνεται με βάση την κατανάλωση ενέργειας, την επαναληψιμότητα, δηλαδή την ικανότητα του να παράγει το ίδιο αποτέλεσμα ανεξαρτήτως επαναλήψεων, τη συμβατότητα, την ευαισθησία, την ακρίβεια, και την ισχύ. Οι Ag-IoT sensors διακρίνονται στις παρακάτω κατηγορίες [4]:

1.3.1 Ακουστικοί

Οι ακουστικοί Ag-IoT sensors έχουν βασική ευθύνη να μετρούν την απόσταση με τη βοήθεια υπερήχων. Μέσω του χρόνου και της ταχύτητας της ηχούς που παράγει το στοιχείο μετρούν την απόσταση. Χαρακτηριστικό παράδειγμα αποτελούν οι σήψεις των φρούτων και των λαχανικών.

1.3.2 Βιολογικοί

Στην κατηγορία αυτή περιλαμβάνονται όλοι οι Ag-IoT sensors που είναι υπεύθυνοι για τη μέτρηση της βιομάζας, της πυκνότητας, την συγκέντρωση χλωροφύλλης κ.α. Η ανάπτυξη του φυτού μπορεί να εκτιμηθεί από τη βιομάζα.

1.3.3 Χημικοί

Οι χημικοί Ag-IoT sensors μπορούν να κατηγοριοποιηθούν σε ηλεκτροχημικούς και φωτοχημικούς. Οι πρώτοι, μετρούν τις ηλεκτρικές μετρικές που γίνονται μέσω χημικών αντιδράσεων ενώ οι δεύτεροι μετρούν την παρουσία χημικών μέσω του φωτός. Επιπλέον, το διοξείδιο του άνθρακα, το οξυγόνο, το μεθάνιο, και η σύσταση του εδάφους είναι κάποια από τα χαρακτηριστικά που μετρούν οι χημικοί Ag-IoT sensors.

1.3.4 Ηλεκτρικοί

Το έργο που επιτελούν αποτελεί η μέτρηση ηλεκτρικών παραμέτρων όπως η ενέργεια, η διαφορά δυναμικού, η ισχύς κ.α. Οι Ag-IoT sensors που είναι υπεύθυνοι για τη μέτρηση της θερμοκρασίας του αέρα και της υγρασίας υπάγονται στη συγκεκριμένη κατηγορία.

1.3.5 Μηχανικοί

Οι μηχανικοί αισθητήρες, μετατρέπουν μία φυσική παράμετρο μέσω ενός μηχανικού συστήματος σε ηλεκτρικό σήμα. Κάποιες από τις παραμέτρους που υπολογίζουν είναι η ταχύτητα, η επιτάχυνση, η πίεση και η απόσταση. Χαρακτηριστικό παράδειγμα αποτελεί το ανεμόμετρο του οποίου είναι υπεύθυνο για τη μέτρηση της ταχύτητας του αέρα. Η υψηλή ταχύτητα του αέρα μπορεί να οδηγήσει στην αλλοίωση των σοδειών και κατ' επέκταση να προκαλέσει πρόβλημα στην παραγωγή των καλλιεργητών.

1.3.6 Οπτικοί

Με την βοήθεια κυμάτων, μετρούν χαρακτηριστικά όπως η κατάσταση των σοδειών βάσει των χρωμάτων. Δεν έχει γίνει ακόμη μεγάλη έρευνα για τους συγκεκριμένους και η χρήση τους είναι περιορισμένη.

1.3.7 Θερμικοί

Οι θερμικοί έχουν μεγάλη χρήση στον τομέα του Ag-IoT. Η μέτρηση της θερμοκρασίας της σκιάς στις σοδειές αποτελεί παράδειγμα αυτού του Ag-IoT sensor. Πραγματοποιείται με τη χρήση της υπέρυθρης ακτινοβολίας και μπορεί να υπολογίσει τη σκιά μέσω του ρυθμού διαπνοής, του δείκτη υδατικής καταπόνησης, και τον ρυθμό άνθησης των φυτών.

1.4 Σκοπός της διπλωματικής

Στόχος της διπλωματικής είναι να εξεταστεί ενδελεχώς η χρήση του Ag-IoT στην καλλιέργεια της ντομάτας και της πιπεριάς. Η πραγμάτωση αυτού του στόχου επιτελείται μέσω των μετρήσεων ορισμένων με τη χρήση των Ag-IoT sensors. Μέσω των μετρήσεων αυτών παράγονται συμπεράσματα για την υγεία της ντομάτας και της πιπεριάς.

Παράλληλα, σκοπός αποτελεί να μπορεί να δημιουργηθεί ένα πρόγραμμα μηχανικής μάθησης που θα κρίνει το ποιες ασθένειες έχει αναπτύξει το φυτό ανάλογα με τις παραμέτρους που μετρήθηκαν. Επιπλέον, θα είναι σε θέση να κρίνει την κατάσταση της υγείας του φυτού βάσει των συνθηκών του περιβάλλοντος που θα διαμορφωθούν τις επόμενες ημέρες.

1.5 Διάρθρωση της διπλωματικής

Στη συνέχεια, παρατίθεται η επεξήγηση των επόμενων κεφαλαίων της εργασίας:

Κεφάλαιο 2

Το κεφάλαιο 2 περιλαμβάνει τη διαδικασία συνδεσμολογίας του κυκλώματος μέτρησης των παραμέτρων. Πιο συγκεκριμένα, περιλαμβάνονται τα χαρακτηριστικά του επεξεργαστή που χρησιμοποιήθηκε, τα χαρακτηριστικά των IoT καθώς και τμήματα του κώδικα με τον οποίο επικοινωνούν με τον υπολογιστή. Επιπλέον, αναλύεται η ασύρματη επικοινωνία του υπολογιστή με τον επεξεργαστή.

Κεφάλαιο 3

Το κεφάλαιο 3 περιλαμβάνει τη διαδικασία μετρήσεων των παραμέτρων της καλλιέργειας της ντομάτας και της πιπεριάς. Επιπλέον, περιλαμβάνει τη διαδικασία της αξιολόγησης της υγείας των φυτών με τη χρήση εφαρμογής μηχανικής μάθησης. Τέλος, παρουσιάζονται τα συμπεράσματα για την υγεία της ντομάτας και της πιπεριάς με βάση την αξιολόγηση των παραμέτρων που μετρήθηκαν.

Κεφάλαιο 4

Το κεφάλαιο 4 σχετίζεται με τη δημιουργία ενός μοντέλου Μηχανικής Μάθησης το οποίο με βάση τις μετρήσεις των προηγούμενων κεφαλαίων, κατηγοριοποιεί τις ασθένειες που μπορεί να αναπτύξει η ντομάτα. Επιπλέον, αποφασίζει τις ασθένειες που είναι πιθανό να αναπτυχθούν τις επόμενες μέρες.

Κεφάλαιο 5

Το κεφάλαιο 5 περιλαμβάνει τα συμπεράσματα από την εκπόνηση της Διπλωματικής Εργασίας , καθώς και μελλοντικές εργασίες βασισμένες σε εκείνη.

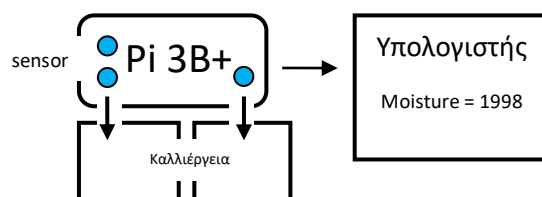
Κεφάλαιο 2

Ανάπτυξη του Πρότυπου Σταθμού Παρακολούθησης και Βελτιστοποίησης Καλλιέργειών

Στο συγκεκριμένο κεφάλαιο, παρουσιάζεται η διαδικασία συνδεσμολογίας του κυκλώματος «Ανάπτυξη του Πρότυπου Σταθμού Παρακολούθησης και Βελτιστοποίησης Καλλιέργειών». Παρουσιάζονται τα εργαλεία που χρησιμοποιήθηκαν, το λογισμικό που χρησιμοποιήθηκε, καθώς και κάποιες βελτιστοποιήσεις του κυκλώματος που σχετίζονται με την ασύρματη επικοινωνία.

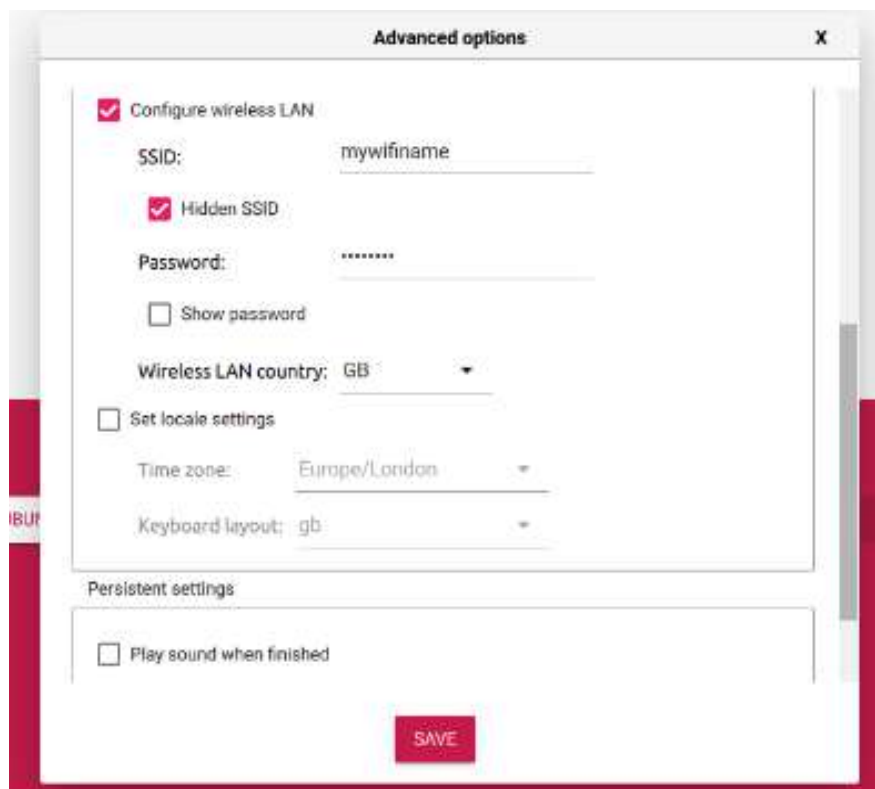
2.1 Διαδικασία συνδεσμολογίας και λήψης λογισμικού του σταθμού

Η παρούσα εργασία έχει στόχο να δημιουργήσει ένα κύκλωμα μέσω του οποίου θα γίνονται μετρήσεις της υγρασίας του εδάφους, της υγρασίας του αέρα, της θερμοκρασίας και της φωτεινής ακτινοβολίας στην ντομάτα και στην πιπεριά κι έπειτα να χρησιμοποιήσει τις μετρήσεις για την ανάλυση των ασθενειών που μπορεί να αναπτύξουν. Επομένως, σημαντική είναι η χρήση κάποιου μικροεπεξεργαστή που θα είναι υπεύθυνος για την υλοποίηση των μετρήσεων. Παράλληλα, εξίσου σημαντική είναι η χρήση των Ag-IoT sensors για την ανίχνευση των χαρακτηριστικών που αναφέρθηκαν και η εύρεση μεθόδων ώστε να βελτιστοποιηθεί το κύκλωμα. Στο παρακάτω σχήμα γίνεται πιο εύληπτη η επικοινωνία μεταξύ του Pi, των Ag-IoT sensors, της καλλιέργειας και του υπολογιστή.



2.1 Δημιουργία σταθμού καλλιέργειών [3]

Η σύνδεση του Pi 3B+ με τον υπολογιστή και το δίκτυο χρησιμοποιεί τόσο κάποιες εφαρμογές όσο και κάποιες τεχνικές διαδικασίες. Βασική προϋπόθεση για την χρήση του Pi είναι η λήψη του λογισμικού Raspbian της εταιρίας Raspberry και η αποθήκευσή του σε SD card. Όταν γίνει η λήψη του λογισμικού, είναι σημαντικό να υλοποιηθούν κάποιες ρυθμίσεις σε σχέση με το προσωπικό όνομα δικτύου και τον προσωπικό κωδικό για την ασφάλεια του χρήστη ως προς την είσοδό του στο desktop του Pi. Μετά την λήψη του λογισμικού πρέπει να γίνουν κάποιες ρυθμίσεις σε σχέση με το δίκτυο που θα συνδεθεί το Pi. Για τον λόγο αυτό, στον φάκελο της SD card δημιουργείται ένα αρχείο με όνομα wpa_supplicant.conf στο οποίο τα χαρακτηριστικά ssid, psk αλλάζουν, ανάλογα τις προδιαγραφές του χρήστη.



```
country=CN
ctrl_interface=DIR=/var/run/wpa_supplicant
GROUP=netdevupdate_config=1
network={
    ssid="WiFi-name"
    psk="WiFi-password"
    key_mgmt=WPA-PSK
    priority=1
}
```

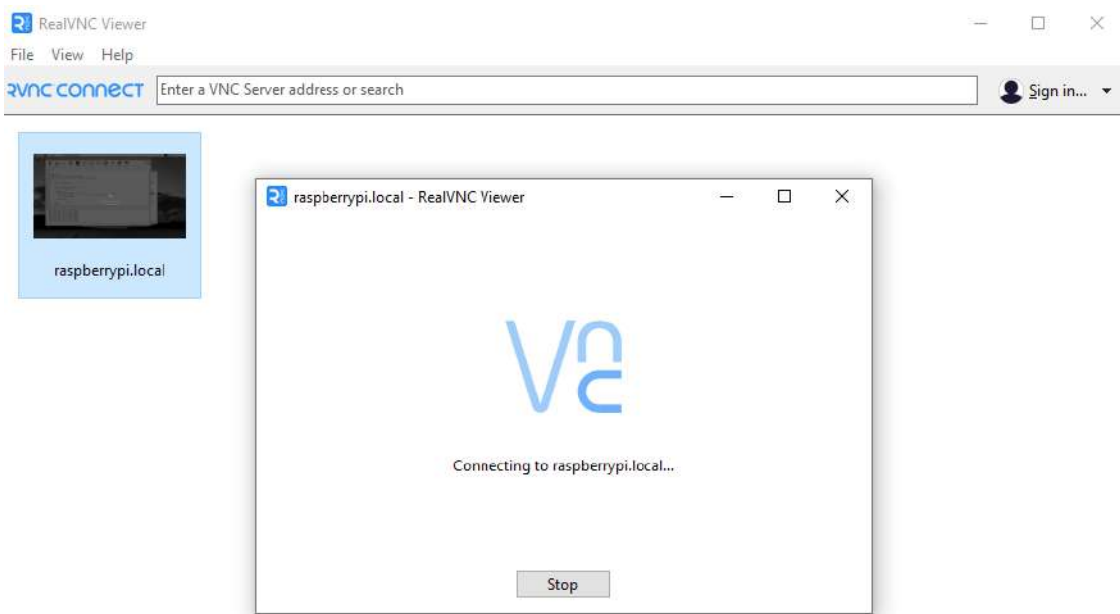
Σχήμα 2.2: Προσθήκη ονόματος χρήστη και κωδικού και δικτύου στο Pi [4]

Στη συνέχεια, τοποθετείται η SD card στο Pi. Το επόμενο βήμα είναι ο υπολογιστής να μπορεί να έχει πρόσβαση στο Desktop του Pi. Για να γίνει αυτό πρέπει να γίνει η σύνδεση του υπολογιστή με το IP Address του το οποίο εφαρμόζεται με τη χρήση της εντολής ssh name@SIP ADDRESS. Στη συνέχεια, παρόλο που η σύνδεση στο Pi έχει γίνει, απαιτείται

χρήση της επιφάνειας εργασίας του Pi η οποία παρέχει πρόσβαση σ' εκείνο με User Interface καθιστώντας πιο εύκολη την υλοποίηση εργασιών, χωρίς την παρουσία εντολών. Αυτό μπορεί να επιτευχθεί με την χρήση της εφαρμογής VNC.

```
CA: Γραμμή εντολών
Microsoft Windows [Version 10.0.19045.3324]
(c) Microsoft Corporation. Με επιφύλαξη κάθε νόμιμου δικαιώματος.
C:\Users\marix> ssh name@IP_
```

Σχήμα 2.3: Προσθήκη της εντολής για την είσοδο στο Pi [5]



Σχήμα 2.4: Χρήση της VNC για την σύνδεση στο Desktop [6]

Αφού γίνει η πρόσβαση στο Desktop του Pi, πρέπει να ληφθεί το λογισμικό το οποίο είναι υπεύθυνο για την εκπόνηση της εργασίας, δηλαδή οι βιβλιοθήκες και το documentation του κάθε Ag-IoT sensor. Η διαδικασία γίνεται μέσω του clone στο repository της εταιρείας SEEDSTUDIO στο github με τις εντολές:

- `curl -sL https://github.com/Seeed-Studio/grove.py/raw/master/install.sh | sudo bash -s -`
- `git clone https://github.com/Seeed-Studio/grove.py`

Μετά την λήψη του απαιτούμενου λογισμικού, στον φάκελο grove.py περιέχεται το documentation του κάθε Ag-IoT sensor σε γλώσσα Python.

Στο παρακάτω σχήμα, τέλος, περιγράφεται συνοπτικά η διαδικασία συνδεσμολογίας και εγκατάστασης του λογισμικού για το κύκλωμα του σταθμού παραγωγής.



Σχήμα 2.5: Διαδικασία δημιουργίας του σταθμού καλλιεργειών [7]



Σχήμα 2.6: Προσθήκη του Ρi στα φυτά για τις μετρήσεις [8]

2.2 Υλικό/Εξαρτήματα και τα χαρακτηριστικά τους

Η λειτουργία του κυκλώματος στηρίζεται στην παροχή συγκεκριμένου Hardware το οποίο πρέπει να πληροί αρκετές προϋποθέσεις. Πιο συγκεκριμένα, όπως προαναφέρθηκε, χρησιμοποιήθηκε το Pi 3B+ ως μικροεπεξεργαστής. Βασικός λόγος της χρήσης του είναι το γεγονός ότι μέσω των GPIO Pins μπορεί και έχει πρόσβαση σε πολλούς Ag-IoT sensors. Επιπλέον, είναι πολύ πιο ταχύς σε σχέση με άλλους επεξεργαστές.

Για την λήψη των Ag-IoT sensors χρησιμοποιήθηκε το Grove Base Kit της εταιρείας seeedstudio. Αποτελεί ένα πακέτο που περιλαμβάνει τους Ag-IoT sensors και το Pi HAT το οποίο περιλαμβάνει πολλές υποδοχές για την άμεση σύνδεση του κάθε Ag-IoT sensor με το Pi. Στα πλαίσια ενός μεγάλου σταθμού παραγωγής η παρουσία πολλών κυκλωμάτων θα δυσχέραινε τις μετρήσεις. Μέσω αυτού αποφεύγεται η δυσκολία στις μετρήσεις, καθώς ο κάθε sensor συνδέεται με ένα καλώδιο στο Pi.

Το κύκλωμα που δημιουργήθηκε επομένως, περιλαμβάνει τα ακόλουθα στοιχεία [8]:

- i. Raspberry Pi 3 Model B+
- ii. Grove Base Hat for Raspberry Pi
- iii. Grove - Moisture Sensor
- iv. Grove - Light Sensor
- v. Grove - Temperature & Humidity Sensor (DHT11)
- vi. SD card
- vii. ESP 8266MOD VENOR

Ακολουθούν οι προδιαγραφές του Pi, του κάθε Ag – IoT sensor και του ESP8266.

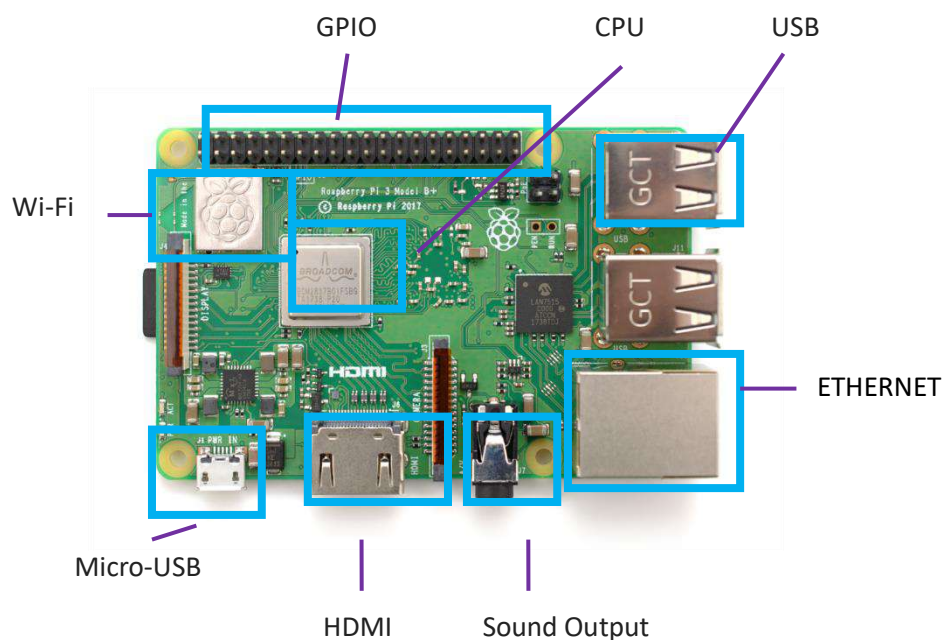
2.2.1 Raspberry Pi 3 Model B+

Το Raspberry Pi 3 Model B+ παρέχει 64 bit και 1.4 GHz τετραπύρρηνο επεξεργαστή, 1 GB RAM, ασύρματο δίκτυο wi-fi που εκπέμπει σε δύο συχνότητες, 2.4 και 5.0 GHz, 802.11 b/g/n/ac wireless LAN, Bluetooth 4.2 και 300 MB/s ethernet.

Διαστάσεις	56.5 mm x 85.6 mm x17 mm
Βάρος	G.W 45 g
Μπαταρία	-
Ισχύς	1.13 A @ 5 Ω
RAM	1 GB DDR2
Core Type	Cortex-A53 64-bit

Πίνακας 2.1: Χαρακτηριστικά του Pi 3B+ [1]

Στο επόμενο σχήμα αναλύονται τα στοιχεία του PI 3B+.



Σχήμα 2.7: Στοιχεία του Pi [9]

Στα πλαίσια της εργασίας χρησιμοποιήθηκαν το Ethernet για την παροχή δικτύου, το micro-usb για την σύνδεση με το ρεύμα, τα GPIO PINS για την σύνδεσή του Pi με το HAT ,το USB για την σύνδεση του Pi με ένα wi-fi module το ESP 8266.

Το Pi 3B+ έχει χρησιμοποιήσει τον ARM A53 Cortex processor. Ο συγκεκριμένος, αποτελείται από 8 επίπεδα Pipeline και 4 cores γεγονός που του επιτρέπει να επιτελεί πολλές ενέργειες παράλληλα. Το γεγονός αυτό είναι πολύ χρήσιμο για τον σταθμό καλλιεργείων, καθώς μπορούν και μετρώνται πολλές παράμετροι μαζί χωρίς την αναμονή για την κάθε μέτρηση [10] .

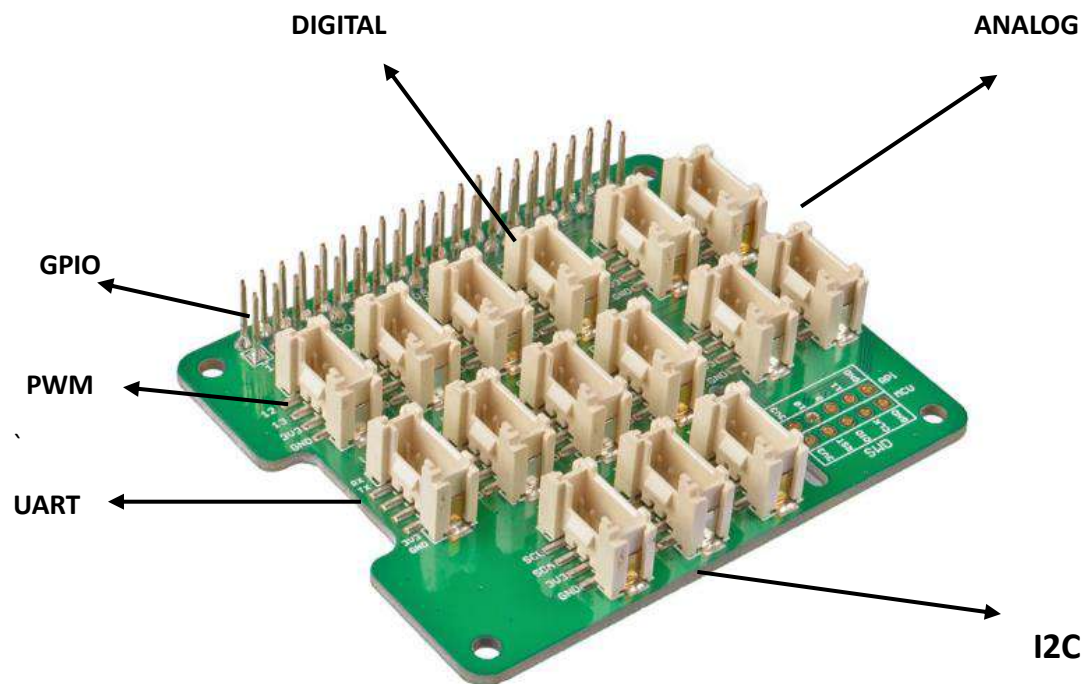
2.2.2 Grove Base Hat for Raspberry Pi

Είναι υπεύθυνο για την παροχή 15 θυρών με τις οποίες μπορεί να συνδεθεί το Pi με τους Ag-IoT sensors. Επιπλέον, περιέχει 24 GPIO Pins και βασίζεται στο MM32 chip. Στον πίνακα που ακολουθεί αναλύονται τα στοιχεία του [8].

Διαστάσεις	55 mm x 65 mm x 20 mm
Βάρος	G.W 24.5 g
Μπαταρία	-
Τάση	3.3 V
Ρεύμα	35 mA
MCU	MM32F031F6P6

Πίνακας 2.2: Χαρακτηριστικά του Grove Base Hat for Raspberry Pi [2]

Στο σχήμα αναλύεται η χρήση της κάθε υποδοχής.



Σχήμα 2.8: Στοιχεία του Pi HAT [10]

Οι υποδοχές χωρίζονται σε GPIO, PWM, UART, ANALOG, I2C, DIGITAL.

GPIO: Είναι υπεύθυνο για την διαχείριση σημάτων εισόδου-εξόδου και για την μετάδοση των δεδομένων στην CPU.

PWM: Η υποδοχή μέσω της οποίας χρησιμοποιείται η διαμόρφωση παλμού (Pulse Width Modulation). Είναι υπεύθυνο για την παραγωγή αναλογικού αποτελέσματος με ψηφιακή είσοδο. Η πληροφορία βρίσκεται στο μήκος της κυματομορφής, που έχει σχήμα παλμού.

I2C: Είναι υπεύθυνο για την επικοινωνία μεταξύ της slave και master συσκευής.

UART: Είναι υπεύθυνο για την μετάδοση της πληροφορίας χωρίς να υπάρχει σήμα ρολογιού.

ANALOG: Έχει ως είσοδο αναλογικά σήματα.

DIGITAL: Έχει ως είσοδο ψηφιακά σήματα.

2.2.3 Grove – Moisture Sensor

Ο Grove - Moisture Sensor είναι υπεύθυνος για την μέτρηση της υγρασίας του εδάφους. Το αποτέλεσμα της μέτρησης που υλοποιεί αποτελεί το ποσοστό της υγρασίας που έχει το έδαφος και την ένδειξη για το αν είναι υγρό, νωπό ή όχι. Συνδέεται σε analog υποδοχή [8]. Στον πίνακα [3] αναφέρονται οι προδιαγραφές του.

Διαστάσεις	60 mm x 20 mm x 6.35 mm
Βάρος	G.W 10 g
Μπαταρία	-
Τάση	3.5-5 V
Ρεύμα	35 mA
Εύρος ενδείξεων για ξηρό έδαφος	0-300
Εύρος ενδείξεων για υγρό έδαφος	300-700
Εύρος ενδείξεων για νερό	700-950
Μέγεθος PCB	2.0 cm x 6.0 cm

Πίνακας 2.3: Χαρακτηριστικά του Grove-Moisture Sensor [3]

2.2.4 Grove - Light Sensor

Είναι υπεύθυνο για την μέτρηση της ποσότητας του φωτός που εισέρχεται στο φυτό. Είναι αναλογική μονάδα και έχει ενσωματώσει μια φωτοαντίσταση LS06-S καθώς και υψηλής ευαισθησίας φωτοδίοδο. Συνδέεται σε analog υποδοχή [8]. Στον πίνακα 2.5 περιγράφονται τα χαρακτηριστικά του.

Διαστάσεις	20mm x20mm x18mm
Βάρος	G.W 8g
Μπαταρία	-
Τάση	3-5V
Ρεύμα	0.5-3 mA
Χρόνος ένδειξης	20-30 ms
Μήκος κύματος	540 nm
Φωτοτρίοδος	GL5528
Input Interface	Analog

Πίνακας 2.5: Χαρακτηριστικά του Grove - Light Sensor [5]

2.2.5 Grove - Temperature & Humidity Sensor (DHT11)

Η λειτουργία του στηρίζεται στη μέτρηση θερμοκρασίας και υγρασίας του αέρα. Είναι Ag-IoT sensor χωρητικότητας. Συνδέεται σε digital υποδοχή [8]. Οι προδιαγραφές του παρουσιάζονται στον πίνακα :

Διαστάσεις	40 mm x 20 mm x 8 mm
Βάρος	G.W 10 g
Μπαταρία	-
Τάση	3.3 V & 5 V
Ρεύμα	1.3 -2.1 mA
Εύρος ενδείξεων υγρασίας	5%-95%
Εύρος ενδείξεων θερμοκρασίας	-20 – 60 °C

Πίνακας 2.4: Χαρακτηριστικά του Temperature & Humidity Sensor [4]

2.2.6 SD card

Η SD card είναι υπεύθυνη για την αποθήκευση ορισμένων προδιαγραφών που σχετίζονται με το λειτουργικό του Pi το Raspbian. Αφού γίνει το download του Desktop

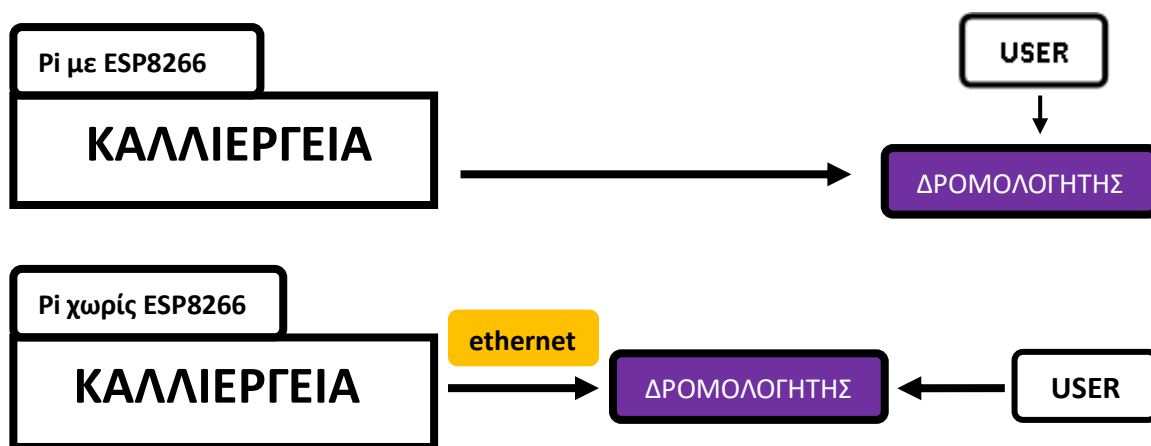
Imager στην sd, τοποθετείται στο Pi με αποτέλεσμα να μπορεί ο χρήστης να χρησιμοποιήσει το desktop του.

2.2.7 ESP8266MOD VENOR

Το Pi 3B+ διαθέτει Wi-Fi. Παρόλα αυτά, πολλές φορές είναι πιθανό να μην μπορεί να εντοπιστεί και να μην του δοθεί κάποιο IP Address. Αυτό μπορεί να συμβεί σε περιπτώσεις που οι εντολές που έχουν δοθεί στην 2.1 να μην γίνονται αποδεκτές από ορισμένους υπολογιστές. Τη λύση στο πρόβλημα μπορεί να δώσει το ESP 8266, καθώς εντοπίζεται κατευθείαν και συνδέεται μέσω USB. Άλλη λύση στο πρόβλημα είναι η χρήση του ethernet. Παρόλα αυτά υπάρχει περιορισμός στην απόσταση του Pi από τον δρομολογητή. Το ESP 8266 επομένως, μετατρέπει την επικοινωνία του Pi με το δίκτυο σε ασύρματη χωρίς τον περιορισμό της απόστασης. Το ESP8266MOD VENOR έχει τα στοιχεία που ακολουθούν [9].

Διαστάσεις	24 mm x 16 mm x 3 mm
Τάση	3.3 V
Digital I/O Pins (DIO)	16
Clock Speed	80 MHz

Πίνακας 2.7: Χαρακτηριστικά του ESP8266MOD VENOR [7]



Σχήμα 2.8: Σύνδεση του Pi με ESP και με Ethernet [10]

2.3 Λογισμικό του Σταθμού Παρακολούθησης

Ο σταθμός παρακολούθησης όπως αναφέρθηκε, χρησιμοποιεί συγκεκριμένο λογισμικό για την επικοινωνία του Pi με τους Ag-IoT sensors. Ο κάθε Ag-IoT sensor χρησιμοποιεί ένα σύνολο βιβλιοθηκών και συναρτήσεων [6].

2.3.1 Λογισμικό του Grove - Temperature & Humidity Sensor

Παρακάτω αναλύεται ο κώδικας που χρησιμοποιήθηκε για την επικοινωνία του Ag-IoT sensor Grove - Temperature & Humidity Sensor με τον υπολογιστή.

```
1#!/usr/bin/env python3
2.
3.import time
4.
5.from seeed_dht import DHT
6.from grove.display.jhd1802 import JHD1802
7.
8.def main():
9.     # Grove - 16x2 LCD(White on Blue) connected to I2C port
11.     lcd = JHD1802()
12.
13.     # Grove - Temperature&Humidity Sensor connected to port D5
14.
15.     sensor = DHT('11', 5)
16.
17.     while True:
18.         humi, temp = sensor.read()
19.         print('temperature {}C, humidity {}'.format(temp,
20.             humi))
21.
22.         lcd.setCursor(0, 0)
23.         lcd.write('temperature: {}'.format(temp))
24.
25.
26.         lcd.setCursor(1, 0)
27.         lcd.write('hum idity: {}'.format(humi))
28.
29.         time.sleep(1)
30.
31.if __name__ == '__main__':
32.     main()
```

Σχήμα 2.9: Λογισμικό του Grove - Temperature & Humidity Sensor [11]

Στην γραμμή 3 έχει χρησιμοποιηθεί η βιβλιοθήκη `seeed_dht`. Η βιβλιοθήκη έχει ως βασικό στόχο να συνδέσει τον Grove - Temperature & Humidity Sensor με το Pi Hat. Η ανάλυση κάποιων σημείων του `documentation` της ίδιας της βιβλιοθήκης, είναι παρακάτω.

```
1. # we must check the device busy flag, bit[7] : 1 for
2.   busy ,0 for idle.
3. while((self._dht10_read_status() & 0x80) != 0):
4.     time.sleep(.5)
5.     print("wait for device not busy")
```

Σχήμα 2.10: `seeed_dht` [12]

Αν η συνθήκη στην `while` είναι διάφορη του μηδενός η συσκευή είναι σε ισχύ.

2.3.2 Λογισμικό του Grove - Light Sensor

Παρακάτω αναλύεται ο κώδικας που έχει χρησιμοποιηθεί για τον Ag-IoT sensor Grove - Light Sensor .

```
1. #!/usr/bin/env python
2.
3. import time
4.
5. from grove.grove_servo import GroveServo
6. from grove.grove_light_sensor_v1_2 import GroveLightSensor
7.
8. def main():
9.     # Grove - Servo connected to PWM port
10.    servo = GroveServo(12)
11.    # Grove - Light Sensor connected to port A0
12.    sensor = GroveLightSensor(0)
13.
14.
15.    while True:
16.        angle = sensor.light * 180 / 1000
17.        print('light value {}, turn to {}
18.degree.'.format(sensor.light, angle))
19.        servo.setAngle(angle)
20.
21.        time.sleep(1)
22.
23.if __name__ == '__main__':
24.    main()
```

Σχήμα 2.11: Λογισμικό του Grove - Light Sensor [13]

Στη γραμμή 12, καλείται η συνάρτηση `GroveLightSensor` για την λειτουργία του Ag-IoT Light sensor. Η συνάρτηση αυτή βρίσκεται στην βιβλιοθήκη `grove.grove_light_sensor_v_2`.

2.3.3 Λογισμικό του Grove-Moisture Sensor

Παρακάτω, αναλύεται ο κώδικας που χρησιμοποιήθηκε για την επικοινωνία του Ag-IoT sensor Grove-Moisture Sensor με τον υπολογιστή. Το λογισμικό του είναι το ακόλουθο:

```
#!/usr/bin/env python

1.
2. import time
3. from mraa import getGpioLookup
4.
5. from grove.grove_moisture_sensor import GroveMoistureSensor
6.
7. def main():
8.
9. # Grove - Moisture Sensor connected to port A0
10. sensor = GroveMoistureSensor(0)
11. while True:
12. mois = sensor.moisture
13. if 0 <= mois and mois < 300:
14. level = 'dry'
15. elif 300 <= mois and mois < 600:
16. level = 'moist'
17. else:
18. level = 'wet'
19. print('moisture: {}, {}'.format(mois, level))
20. time.sleep(1)
21. if __name__ == '__main__':
22. main()
```

Σχήμα 2.12: Λογισμικό του Grove-Moisture Sensor [14]

Στη γραμμή 6, η βιβλιοθήκη `grove.grove_moisture_sensor` σχετίζεται με τη λειτουργία του moisture sensor. Η βιβλιοθήκη περιέχει:

```
1. print('Detecting moisture...')
2. while True:
3. m = sensor.moisture
4. if 0 <= m and m < 300:
5.     result = 'Dry'
6. elif 300 <= m and m < 600:
7.     result = 'Moist'
8. else:
9. result = 'Wet'
10. print('Moisture value: {}, {}'.format(m, result))
```

Σχήμα 2.13: `grove_moisture_sensor` [15]

Μέσω της `while` ορίζονται οι περιοχές που το φυτό είναι στεγνό, `[0, 300]`, νωπό `[300, 600]` και υγρό αν είναι πιο μεγάλο του 600.

2.4 Πλεονεκτήματα και μειονεκτήματα της χρήσης του Pi 3B+

Το Pi 3B+ έχει χρησιμοποιηθεί για την εκπόνηση της εργασίας. Η χρήση του, χαρακτηρίζεται από κάποια μειονεκτήματα και κάποια πλεονεκτήματα, σε σχέση τόσο με άλλους επεξεργαστές όσο και με τα υπόλοιπα Pi.

Τα πλεονεκτήματα της χρήσης του Pi 3B+ και γενικότερα ενός Pi είναι ποικίλα σε σχέση με άλλους επεξεργαστές όπως το Arduino. Το Pi, διαθέτει επεξεργαστή στα 1.6 GHz που είναι σαφώς πιο γρήγορος, ευνοώντας την πιο αποδοτική παραγωγή. Επιπλέον, το Pi περιλαμβάνει τα GPIO Pins μέσω των οποίων μπορούν να δημιουργηθούν πολλά κυκλώματα σε αντίθεση με άλλους επεξεργαστές που δεν διαθέτουν αυτό το προνόμιο [4]. Το Pi 3B+ συγκεκριμένα, σε σχέση με προηγούμενες εκδόσεις του Pi όπως το 0, A, B και 2 υπερτερεί στο ότι περιλαμβάνει Ethernet, Bluetooth και Wi-Fi [5]. Επιπλέον, αποτελείται από 40 GPIO Pins σε αντίθεση με τις προηγούμενες που έχουν 26 [5]. Σε σχέση με το Pi 4B, το 3B+ έχει λιγότερες απαιτήσεις σε ισχύ. Επιπρόσθετα, χρησιμοποιεί HDMI καλώδιο σε αντίθεση με το 4 που χρησιμοποιεί micro HDMI, το οποίο δεν συναντάται τόσο συχνά [7].

Το Pi έχει και κάποια μειονεκτήματα σε σχέση με άλλους μικροεπεξεργαστές. Δεν διαθέτει δικό του χώρο αποθήκευσης με αποτέλεσμα να χρησιμοποιηθεί SD Card η οποία είναι πιο αργή. Επιπλέον, δεν χρησιμοποιεί το λειτουργικό Windows, το οποίο σε σχέση με το Linux χάρη στο User Interface είναι πιο εύκολο στην χρήση [4]. Η RAM είναι μικρότερη από εκείνη του 4B+ με αποτέλεσμα οι εργασίες να γίνονται σε πολύ περισσότερο χρόνο. Τέλος, το 4B έχει περισσότερες θύρες από το 3B+ και έχει υψηλό CPU clock (1.5GHz) και πιο ισχυρό επεξεργαστή τον Cortex-A72 (ARM v8) 64-bit [7].

Κεφάλαιο 3

Μετρούμενα Χαρακτηριστικά Καλλιεργειών

3.1 Ανάλυση των μετρούμενων παραμέτρων και σκοπός της μέτρησης

Η εκπόνηση της εργασίας έχει ως βασικό στόχο να μελετήσει τις ασθένειες που μπορούν να εκδηλώσουν η ντομάτα και η πιπεριά με τη χρήση συγκεκριμένων εργαλείων που αναφέρθηκαν στην προηγούμενη ενότητα. Η μέτρηση επιλεγμένων παραμέτρων όπως η θερμοκρασία, η υγρασία του εδάφους, η υγρασία του αέρα και η ακτινοβολία του φωτός έχουν καθοριστική σημασία στην πρόβλεψη της υγείας των φυτών. Στα πλαίσια της εργασίας πραγματοποιήθηκαν μετρήσεις των συγκεκριμένων παραμέτρων για περίπου 3 μήνες. Οι μετρήσεις πραγματοποιούνταν ανά 4-5 ημέρες γιατί κατά μέσο όρο τα φυτά εκδηλώνουν ασθένειες σε αυτό το διάστημα αφού εκτεθούν σε κάποια καιρική συνθήκη [12]. Μετά τη μέτρηση των παραμέτρων, ακολουθούσε η λήψη φωτογραφιών των φύλλων των φυτών και η είσοδός τους σε ένα πρόγραμμα Μηχανικής Μάθησης για την αξιολόγηση της ασθένειας του φυτού. Έτσι, η κάθε μέτρηση έχει αντιστοίχιση με μία ή και παραπάνω ασθένειες. Βασικός στόχος της λήψης των μετρήσεων και της χρήσης του προγράμματος Μηχανικής Μάθησης είναι να βγουν συμπεράσματα σε σχέση με το ποιες συνθήκες συσχετίζονται με την κάθε ασθένεια. Όπως έχει αναλυθεί και στο 1^ο κεφάλαιο, οι καλλιεργητές για την αποφυγή ασθενειών ή και την αντιμετώπισή τους χρησιμοποιούσαν την εμπειρία. Οι γνώσεις τους αυτές τους βοηθούν να αντιμετωπίσουν τις ασθένειες που προκύπτουν στις καλλιέργειες τους αλλά δεν τους βοηθά στην πρόβλεψη της εκδήλωσης αυτών των ασθενειών. Η χρήση στοιχείων μετρήσεων επομένως, σε συνδυασμό με το επόμενο κεφάλαιο θα μπορεί να προειδοποιεί τους καλλιεργητές για την επερχόμενη ασθένεια που θα αναπτύξει το φυτό τους.

Η κάθε μέτρηση επομένως, έχει την ακόλουθη μορφή:

Light	temperature	humidity	moisture	Health
699	30	48	1839	bacterial spot
538	26	60	1817	leaf mold
698	26	52	1875	Yellow leaf curl

Σχήμα 3.1 : Μετρήσεις παραμέτρων και αξιολόγηση της υγείας της ντομάτας [16]

Τα αποτελέσματα των μετρήσεων αναπαριστώνται στο documentation του κάθε Ag-IoT.

```

MOISTURE
moisture: 2302, wet
moisture: 2306, wet

TEMPERATURE, HUMIDITY
Temperature 24C, humidity 43%

LIGHT
light value 698, turn to 126.36 degree.
light value 698, turn to 125.82 degree.

```

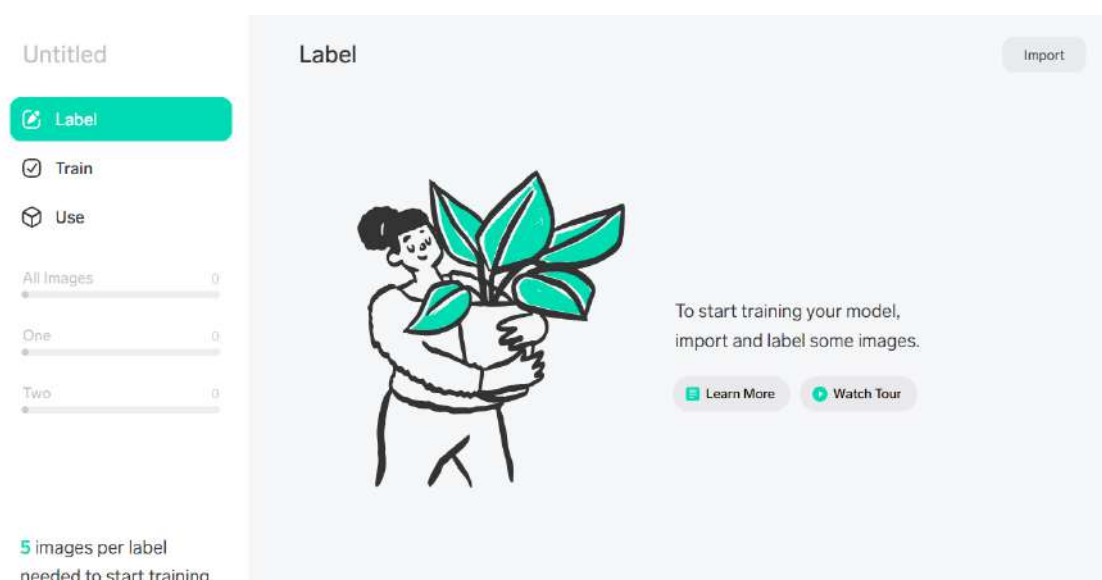
Σχήμα 3.2: Αποτελέσματα των μετρήσεων [17]

Οι μονάδες μέτρησης που χρησιμοποιήθηκαν είναι «%» για την υγρασία του αέρα, «° C» για την θερμοκρασία, ενώ για την υγρασία του εδάφους και την ακτινοβολία του φωτός δεν χρησιμοποιήθηκαν μονάδες μέτρησης.

Επομένως, οι παράμετροι σε μορφή διαστημάτων θα έχουν την ικανότητα να θέτουν τα όρια που μπορεί να αναπτυχθεί μια ασθένεια. Πολλές φορές μια παράμετρος μπορεί να μην έχει καμία σχέση με κάποια ασθένεια και οπότε να μην χρειάζεται για τη λήψη της απόφασης. Το διάστημα που θα προκύψει για την κάθε παράμετρο ανάλογα την ασθένεια θα αναλυθεί στις επόμενες ενότητες.

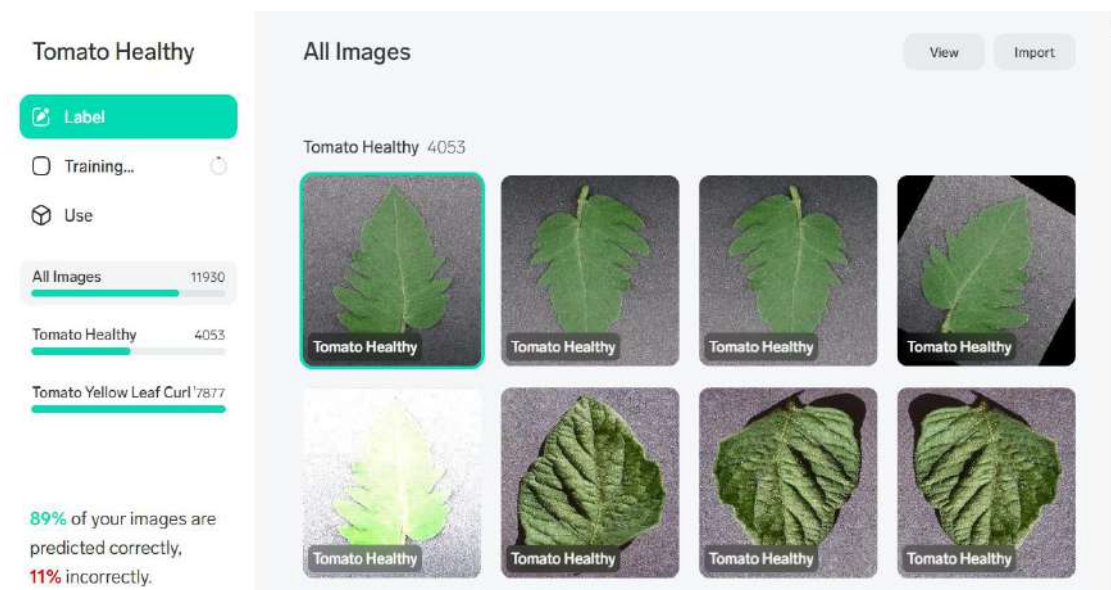
3.2 Πρόγραμμα Μηχανικής Μάθησης για την αξιολόγηση της υγείας των φυτών

Η αξιολόγηση της κατάστασης των φυτών στηρίχθηκε στην εφαρμογή Μηχανικής Μάθησης Lobe [19]. Μετά τις μετρήσεις των παραμέτρων που έχουν αναφερθεί, φωτογραφίες από τα φύλλα των φυτών εισάγονται στο συγκεκριμένο πρόγραμμα, κι έπειτα εκείνο εμφανίζει την κατάσταση της υγείας τους ως αποτέλεσμα. Η διαδικασία αυτή υλοποιείται σε 3 επίπεδα. Το “IMPORT” είναι η διαδικασία με την οποία γίνεται η προσθήκη των Datasets στο πρόγραμμα. Για την πιπεριά χρησιμοποιήθηκε το dataset από τη σελίδα Kaggle και διαχωρίζει το αν είναι υγιής ή αν έχει bacterial spots. Στην περίπτωση της ντομάτας, το πρόγραμμα αποφασίζει για το ποιες ασθένειες έχει εμφανίσει. Λόγω του μεγέθους του Dataset δεν ήταν εύκολο να χρησιμοποιηθεί ένα project με 10 ασθένειες που περιέχονταν. Για αυτό, διαχωρίστηκε σε 9 projects στα οποία οι επιλογές ήταν ανάμεσα στο “HEALTHY” και “DISEASE”.



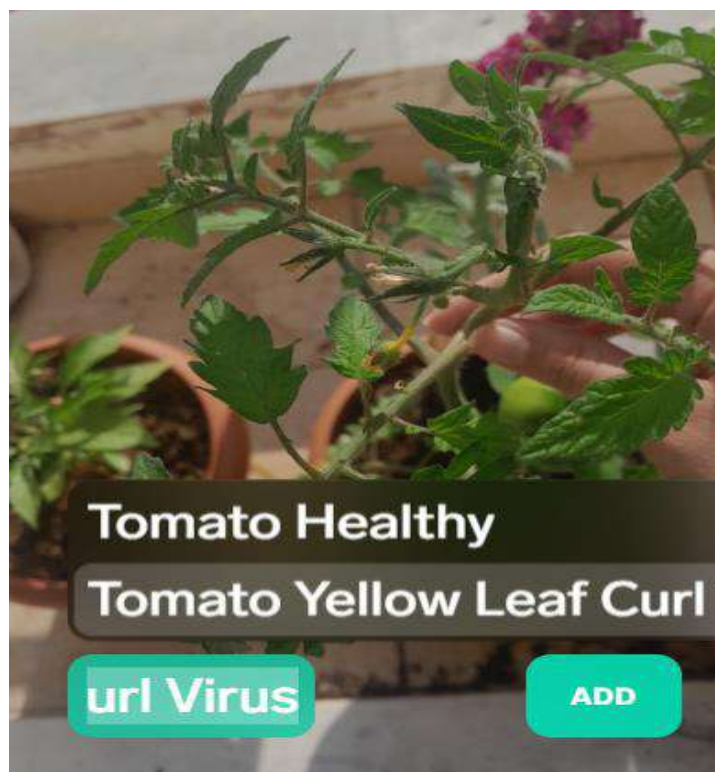
Σχήμα 3.3: Διαδικασία του Import [18]

Στη συνέχεια, πραγματοποιείται η διαδικασία της εκπαίδευσης. Η εκπαίδευση του προγράμματος ολοκληρώνεται σε αρκετά μεγάλο διάστημα, 20 – 40 λεπτά. Αυτό οφείλεται στο γεγονός ότι το πρόγραμμα ελέγχει όλες τις φωτογραφίες για το αν πληρούν συγκεκριμένες προϋποθέσεις όπως η ευκρίνεια και το μέγεθος. Όταν ολοκληρωθεί η διαδικασία, το πρόγραμμα είναι σε θέση να εντοπίσει αν η φωτογραφία «ταιριάζει» με το dataset που έχει χρησιμοποιηθεί.



Σχήμα 3.4: Διαδικασία του Training [19]

Τέλος η διαδικασία του “USE” περιλαμβάνει την προσθήκη της φωτογραφίας με την οποία γίνεται εύληπτη η κατάσταση της υγείας του φυτού και το πρόγραμμα βγάζει ως αποτέλεσμα το αν είναι υγιές ή την ασθένεια που μπορεί να έχει. Πολλές φορές δεν υπάρχει πλήρης ταύτιση της φωτογραφίας που έχει εισαχθεί με το dataset που χρησιμοποιήθηκε. Για τον λόγο αυτό όταν εμφανίζεται το αποτέλεσμα, το παραλληλόγραμμο στο οποίο βρίσκεται το όνομα της ασθένειας δεν είναι πλήρως χρωματισμένο.



Σχήμα 3.5: Διαδικασία του Use [20]

3.3 Συμπεράσματα για την υγεία της πιπεριάς

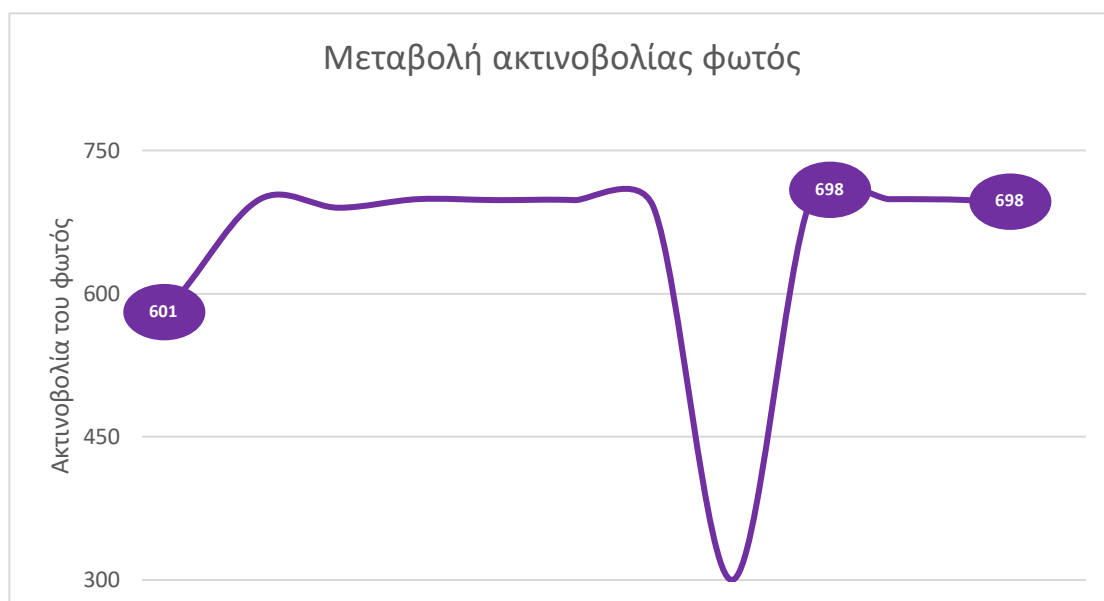
Για την αξιολόγηση της υγείας της πιπεριάς χρησιμοποιήθηκε ένα έγκυρο dataset από την εφαρμογή Kaggle. Το dataset που χρησιμοποιήθηκε έχει διαχωρίσει την πιπεριά σε «υγιή» και «μη υγιή» ανάλογα με την παρουσία σημαδιών βακτηρίων (bacterial spots). Στην πιπεριά, με βάση τις μετρήσεις που έχουν υλοποιηθεί, η υγιής χρειάζεται σε μεγάλο βαθμό την παρουσία νερού. Παρατηρήθηκε από τις μετρήσεις ότι σε ημέρες με σχετικά υψηλή υγρασία αέρα, εδάφους και μέση θερμοκρασία η πιπεριά είχε λιγότερα έως και μηδενικά σημάδια από βακτήρια. Από την άλλη, σε περιόδους με λιγότερη υγρασία και υψηλές θερμοκρασίες παρουσιάζονται κίτρινες κηλίδες, το ωίδιο. Επιπλέον, όταν τα επίπεδα νερού ξεπερνούν ένα συγκεκριμένο όριο, η πιπεριά χαρακτηρίζεται από σημάδια βακτηρίων που είναι μαύρες κηλίδες.

Με βάση την ανάλυση των μετρήσεων, οι ιδανικές θερμοκρασίες της καλλιέργειας είναι στο διάστημα 18 - 30 °C με βασική προϋπόθεση η υγρασία αέρα να είναι σε μέσα προς υψηλά επίπεδα, δηλαδή 55 – 65% και η υγρασία του εδάφους να είναι στο διάστημα 1890 – 2090. Από την άλλη, το ωίδιο εμφανίζεται σε υψηλές θερμοκρασίες, δηλαδή μεγαλύτερη από 26°C και χαμηλά ποσοστά νερού δηλαδή λιγότερο από 55% και 1890 αντίστοιχα. Τα σημάδια βακτηρίων εμφανίζονται σε υψηλά ποσοστά νερού δηλαδή σε διαστήματα που ξεπερνούν τα άνω όρια των προηγούμενων διαστημάτων. Στα παρακάτω σχήματα γίνεται εμφανής η διαφοροποίηση μεταξύ των ασθενειών της πιπεριάς.



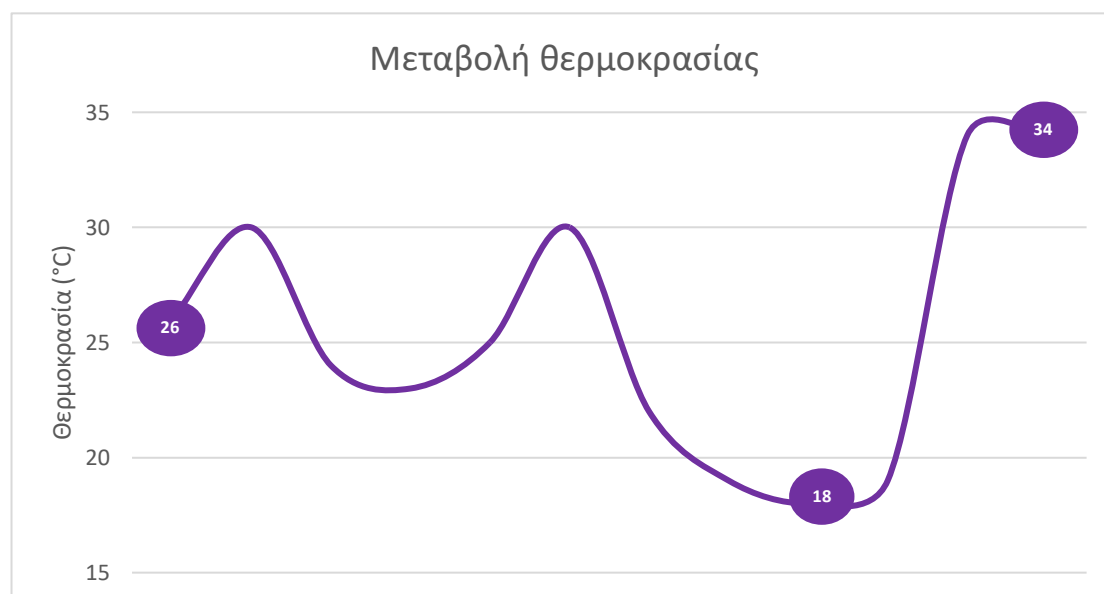
Σχήμα 3.6: Ασθένειες πιπεριάς: Μαύρο σημάδι και ωίδιο [21]

Στα παρακάτω διαγράμματα αναλύονται οι μεταβολές στην υγεία του σε σχέση με τις συνθήκες υγρασίας, θερμοκρασίας και φωτεινής ακτινοβολίας.



Σχήμα 3.7: Μεταβολή της ακτινοβολίας του φωτός σε σχέση με τις ημέρες [22]

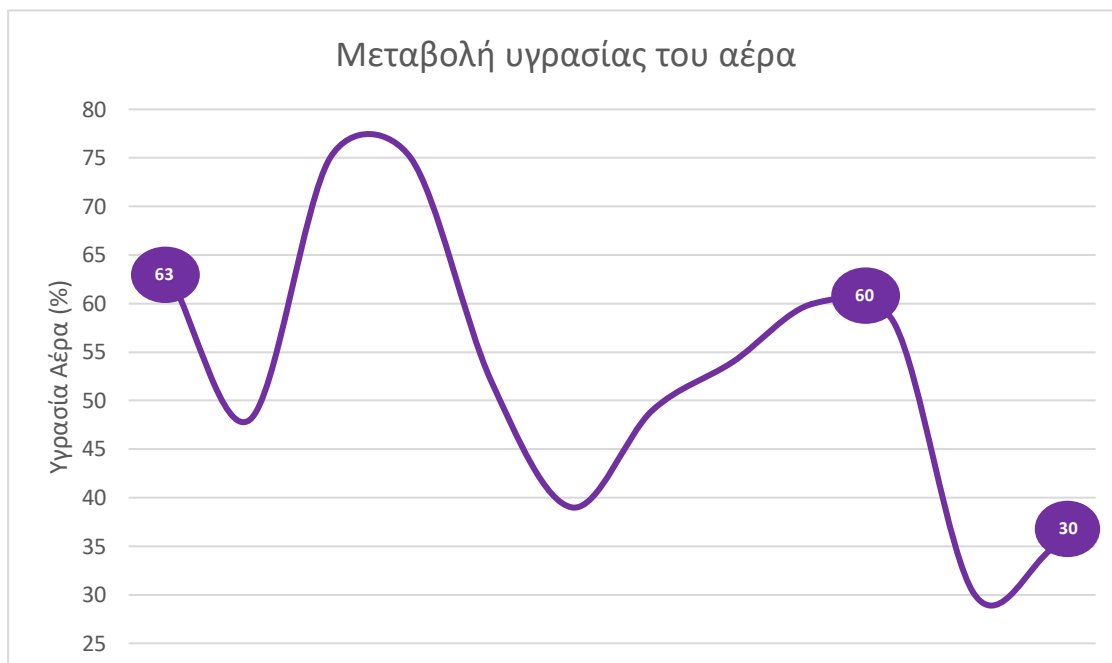
Στο προηγούμενο διάγραμμα, έχουν σηματοδοτηθεί οι περίοδοι στις οποίες το φυτό είχε πλήρη υγεία. Σε σχέση με τις υπόλοιπες τιμές της μεταβλητής, δεν παρουσιάζεται μεγάλη διαφοροποίηση. Οι τιμές συνήθως κυμαίνονται στο διάστημα [600, 700]. Βέλτιστη υγεία παρουσίασε με τις τιμές 601, 698 και 698. Οι μετρήσεις έγιναν την περίοδο της Άνοιξης και του Καλοκαιριού καθιστώντας την ηλιακή ακτινοβολία σχεδόν αμετάβλητη. Επομένως, τις περιόδους αυτές η ηλιακή ακτινοβολία δεν είχε καθοριστικό ρόλο στην εξέλιξη της υγείας του φυτού.



Σχήμα 3.8: Μεταβολή της θερμοκρασίας σε σχέση με τις ημέρες [23]

Με βάση το σχήμα, παρατηρείται ότι το φυτό παρουσιάζει βέλτιστη υγεία στις θερμοκρασίες 18, 25, και 34 °C. Αυτές οι θερμοκρασίες έχουν μια απόσταση μεταξύ τους καθιστώντας δύσκολη την ανάλυση για την εύρεση της βέλτιστης θερμοκρασίας. Παρόλα

αυτά, όπως θα φανεί στη συνέχεια στη θερμοκρασία 18 °C η υγρασία είναι σε χαμηλό βαθμό σε σχέση με τις άλλες θερμοκρασίες και έτσι το φυτό χαρακτηρίζεται από νωπό έδαφος το οποίο όπως αναφέρθηκε είναι το ιδανικό για την καλλιέργεια της πιπεριάς.



Σχήμα 3.9: Μεταβολή της υγρασίας του αέρα σε σχέση με τις ημέρες [24]

Η μεταβολή της υγρασίας του αέρα περιλαμβάνει τιμές στο διάστημα 30 – 75%. Το φυτό παρουσιάζει βέλτιστη υγεία σε μέσες προς μεγάλες τιμές, δηλαδή 60 και 63%. Παρόλα αυτά, χαρακτηρίζεται και ως υγιές σε περίοδο που έχει 30% υγρασία. Ο λόγος που συμβαίνει αυτό είναι γιατί εκείνη την ημέρα η υγρασία του εδάφους είναι μεγάλη γεγονός που οδηγεί σε νωπό έδαφος. Επιπλέον, το φυτό δεν παρουσιάζει ασθένειες στα τοπικά μέγιστα του διαγράμματος. Αυτό σημαίνει ότι το φυτό ανακάμπτει σ' εκείνες τις περιοχές ύστερα από περίοδο απουσίας νερού.



Σχήμα 3.10: Μεταβολή υγρασίας του εδάφους σε σχέση με τις ημέρες [25]

Η υγρασία του εδάφους αποτελεί βάσει του διαγράμματος τον παράγοντα που καθορίζει σε μεγαλύτερο βαθμό την υγεία του φυτού. Πιο συγκεκριμένα, βέλτιστη υγεία παρουσιάζει σε πολύ κοντινές τιμές, δηλαδή 1880, 2065, 2090. Το γεγονός αυτό υποδεικνύει το ότι οι μέσες τιμές είναι οι ιδανικές για την καλλιέργεια. Σε πολύ υψηλές και πολύ χαμηλές τιμές, το φυτό παρουσιάζει τις ασθένειες που έχουν αναφερθεί.

Επομένως, το αποτέλεσμα από την ανάλυση της πιπεριάς είναι ότι χρειάζεται ένα σχετικά υγρό περιβάλλον για να αναπτυχθεί με νωπό έδαφος. Η αρκετά αυξημένη θερμοκρασία σε συνδυασμό με χαμηλή υγρασία δημιουργεί ένα περιβάλλον που βοηθά στη δημιουργία ασθενειών. Το ίδιο συμβαίνει και σε περιπτώσεις που η υγρασία είναι αρκετά αυξημένη και η θερμοκρασία είναι χαμηλή.

Έτσι, μπορούν βγουν συμπεράσματα για τις συνθήκες που αντιστοιχούν σε κάθε ασθένεια, όπως στο σχήμα:

	LIGHT	TEMPERATURE	HUMIDITY	MOISTURE
ΑΣΘΕΝΕΙΑ Α	$[a_1, a_2]$	$[\beta_1, \beta_2]$	$[\gamma_1, \gamma_2]$	$[\delta_1, \delta_2]$
ΑΣΘΕΝΕΙΑ Β	$[a_2, a_3]$	$[\beta_2, \beta_3]$	$[\gamma_2, \gamma_3]$	$[\delta_2, \delta_3]$
- ΑΣΘΕΝΕΙΑ Γ	$[a_3, a_4]$	$[\beta_3, \beta_4]$	$[\gamma_3, \gamma_4]$	$[\delta_3, \delta_4]$

	LIGHT	TEMPERATURE	HUMIDITY	MOISTURE
ΥΓΙΗΣ	[601, 698]	18, 26, 34	35, 60, 63	1890, 2060, 2090
ΜΗ ΥΓΙΗΣ	[300, 698]	[19, 34]	[30, 75]	[1430, 2307]

Σχήμα 3.11: Διαχωρισμός ασθενειών [26]

3.4 Συμπεράσματα για την υγεία της ντομάτας

Η υγεία της ντομάτας είχε διαφορετικού είδους ανάλυση. Το dataset που χρησιμοποιήθηκε περιλαμβάνει 10 περιπτώσεις που μπορεί να αναπτύξει η ντομάτα. Οι περιπτώσεις είναι: Yellow leaf curl, Bacterial Spot, Target Spot, Leaf Mold, Spider Mites, Mosaic Virus, Early Blight, Late Blight, Septoria Leaf spot και Healthy. Η πιο συχνή ασθένεια που παρουσίασε η ντομάτα ήταν η Yellow Leaf Curl. Η ασθένεια αυτή εμφανίστηκε σε ποσοστό 60% των μετρήσεων. Η Yellow Leaf Curl οφείλεται στην μεγάλη έκθεση σε υψηλές θερμοκρασίες. Σε περιπτώσεις που το φυτό τοποθετηθεί σε σημείο με αρκετά υψηλές θερμοκρασίες παρόλη την υγρασία που υπάρχει στο έδαφος εμφανίζει αυτή την ασθένεια.

Κατά την εμφάνιση αυτής της ασθένειας, τα φύλλα δημιουργούν μια καμπύλη προς το εσωτερικό έτσι ώστε να περιορίσουν την έκθεσή τους στις περιβαλλοντικές συνθήκες. Στη διάρκεια των μετρήσεων οι θερμοκρασίες ήταν αρκετά υψηλές, 25% – 35% γεγονός που οδήγησε στην ανάπτυξη αυτής της ασθένειας. Στο παρακάτω σχήμα γίνεται πιο εύληπτο το συγκεκριμένο χαρακτηριστικό της :



Σχήμα 3.12: Ντομάτα με και χωρίς Yellow Leaf Curl [27]

Γίνεται πιο εύκολο να κατανοήσει κάποιος ότι στην αριστερή εικόνα, η κλίση των φύλλων είναι προς το εσωτερικό του φυτού με στόχο την προστασία του από τις περιβαλλοντικές συνθήκες. Στη δεξιά το φύλλο είναι υγιές και δεν έχει δημιουργήσει καμία κλίση.

Οι επόμενες πιο συχνές ασθένειες του φυτού ήταν οι Early Blight και Target Spot. Η Early Blight εμφανίζεται κυρίως σε περιπτώσεις που το φύλλο του φυτού έρχεται αντιμέτωπο με μεγάλη ποσότητα υγρασίας. Η υγρασία παίζει αρκετά πρωτεύων ρόλο στην υγεία του φυτού. Παρόλα αυτά, πρέπει το φύλλο να μην έρχεται σε επαφή με μεγάλη ποσότητα νερού [20]. Μόνο το έδαφος είναι σημαντικό να έχει αρκετή υγρασία. Επομένως, η συγκεκριμένη ασθένεια εμφανίζεται σε περιπτώσεις που υπάρχουν πολλές βροχοπτώσεις και αυξημένη υγρασία στο περιβάλλον. Στις περιπτώσεις που εμφανίστηκε αυτή η ασθένεια οι μετρήσεις της ήταν 63% – 75% που είναι αρκετά υψηλές. Η Target Spot αποτελεί αρκετά παρόμοια ασθένεια με την προηγούμενη. Οι ασθένειες διαφέρουν στην εμφάνιση. Η πρώτη, εμφανίζει μεγάλα σημάδια καφέ χρώματος που καταλαμβάνουν μεγάλη επιφάνεια, ενώ η δεύτερη εμφανίζει καφέ σημάδια που καταλαμβάνουν λιγότερη επιφάνεια [11]. Στα παρακάτω σχήματα γίνονται κατανοητές οι συγκεκριμένες ασθένειες.



Σχήμα 3.13: Target Spot και Early Blight [28]

Η επόμενη πιο συχνή ασθένεια που εμφανίστηκε είναι τα Bacterial spots. Η αιτία δημιουργίας τους είναι η υψηλή υγρασία στα φύλλα τους και οφείλεται στην παρουσία βακτηρίων [13]. Στο επόμενο σχήμα παρουσιάζεται η ασθένεια.



Σχήμα 3.14: Bacterial spots [29]

Οι επόμενες ασθένειες που δημιουργήθηκαν εμφανίστηκαν πολύ λιγότερο από τις προηγούμενες. Αυτές είναι οι Septoria Leaf Spot, Late Blight, Mosaic Virus και Leaf Mold. Η πρώτη αναπτύσσεται σε περιοχές που υπάρχει αρκετή υγρασία για πολλές ημέρες και αποτελεί μια από τις πιο καταστροφικές ασθένειες για το φυτό. Τα φύλλα περιέχουν σκούρες κηλίδες με γκρι περιγράμματα [14]. Η υγρασία βρισκόταν σε ποσοστό 75% και 2304. Στην Late Blight οι θερμοκρασίες είναι ήπιες, και υπάρχει αυξημένη υγρασία. Το

φύλλο του φυτού περιέχει αρκετά μεγάλες καφέ κηλίδες [16]. Η επόμενη ασθένεια δεν οφείλεται στην ποσότητα υγρασίας, στη θερμοκρασία ή στην φωτεινή ακτινοβολία. Βασική αιτία είναι ο καπνός και η ένωση των τμημάτων του φυτού σε περίπτωση που κοπεί ή σχηματιστεί ρωγμή [21]. Τέλος, η Leaf mold δημιουργείται από αρκετά υψηλή υγρασία. Περιλαμβάνει καφέ κηλίδες με βασικό γνώρισμα την αλλαγή χρώματος των κηλίδων κατά την εξέλιξη της ασθένειας. Σε πρώτη φάση οι κηλίδες είναι γκρι, ύστερα κίτρινες και μετά καφέ [18].



Σχήμα 3.15: Septoria Leaf Spot, Late Blight [30]



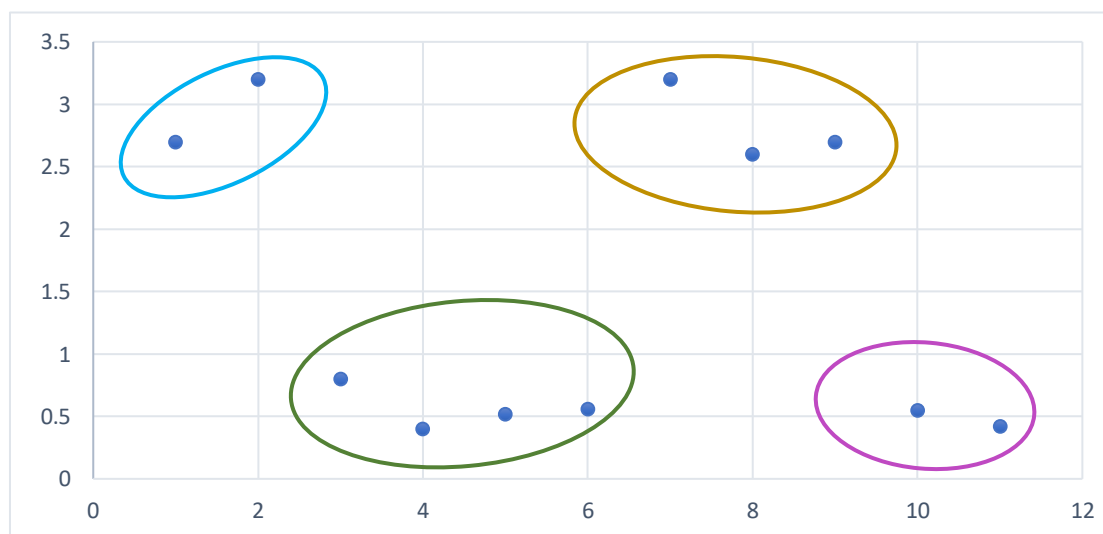
Σχήμα 3.16 Mosaic Virus και Leaf Mold [31]

Η ανάλυση που έγινε στις ασθένειες της ντομάτας δεν επαρκεί για τη δημιουργία διαστημάτων που διαχωρίζουν την κάθε ασθένεια. Το πλήθος των ασθενειών είναι αρκετά μεγάλο. Για το λόγο αυτό στο επόμενο κεφάλαιο δημιουργήθηκε ένα πρόγραμμα Μηχανικής Μάθησης που είναι υπεύθυνο για τη δημιουργία αυτών των διαστημάτων.

Κεφάλαιο 4

Βελτιστοποίηση της καλλιέργειας μέσω δημιουργίας μοντέλου μηχανικής μάθησης για την πρόβλεψη πιθανής ασθένειας

Στο συγκεκριμένο κεφάλαιο, βασικός στόχος αποτελεί να συγκεκριμενοποιηθούν τα διαστήματα των παραμέτρων της υγρασίας του αέρα, της υγρασίας του εδάφους της θερμοκρασίας και της φωτεινής ακτινοβολίας ανάλογα με την ασθένεια. Στο προηγούμενο κεφάλαιο, η ανάλυση της υγείας της πιπεριάς ήταν διαχειρίσιμη, καθώς έπρεπε να βγουν συμπεράσματα ανάμεσα σε δύο καταστάσεις: υγιής και μη υγιής. Για το λόγο αυτό δεν είναι απαραίτητη η δημιουργία κάποιου μοντέλου που να διαχωρίζει τα διαστήματα. Στην περίπτωση της ντομάτας όμως, η επιλογή ανάμεσα σε 10 ασθένειες λαμβάνει υπόψιν πολλές παραμέτρους, τις οποίες ένα πρόγραμμα είναι σε μεγάλο βαθμό ικανό να αποσαφηνίσει. Πιο συγκεκριμένα, στο παρόν κεφάλαιο θα αναπτυχθεί ένα πρόγραμμα μηχανικής μάθησης το οποίο με την μέθοδο του k-means clustering διαχωρίζει τις "περιοχές" που εμφανίζονται οι ασθένειες. Παράλληλα, ο διαχωρισμός των ασθενειών θα χρησιμοποιηθεί για την πρόβλεψη πιθανής ασθένειας που μπορεί να εκδηλωθεί στο μέλλον με τη χρήση του Web Scrapping.



Σχήμα 4.1 k-means clustering [32]

4.1 Επιλογή μοντέλου μηχανικής μάθησης

Το μοντέλο που χρησιμοποιήθηκε ονομάζεται k-means clustering. Το συγκεκριμένο μοντέλο είναι υπεύθυνο στο να διαχωρίσει ένα dataset σε συστάδες (clusters). Ο διαχωρισμός των συστάδων γίνεται μέσω της χρήσης της ευκλείδειας απόστασης. Πιο συγκεκριμένα, επιλέγονται τυχαία στον χώρο κάποια σημεία ανάλογα με τον αριθμό K. Τα σημεία αυτά ονομάζονται κέντρα (centroids). Στη συνέχεια, υπολογίζεται η απόσταση κάθε κέντρου με όλα τα σημεία του dataset. Τα σημεία τα οποία έχουν τις πιο κοντινές αποστάσεις δημιουργούν μια συστάδα. Στη συνέχεια, τη διαδικασία αυτή ακολουθούν οι αλλαγές των κέντρων μέσω της μέτρησης του μέσου όρου των σημείων της κάθε συστάδας. Η διαδικασία γίνεται έως ότου οι συστάδες σταθεροποιηθούν.

Οι μαθηματικές σχέσεις που ακολουθεί το K-MEANS clustering είναι οι ακόλουθες:

- Ευκλείδεια απόσταση για τον υπολογισμό των αποστάσεων μεταξύ των σημείων -datapoints:

$$d(x, y) = \sqrt{\sum_{i=1}^n (x_i - y_i)^2} \quad 4-1$$

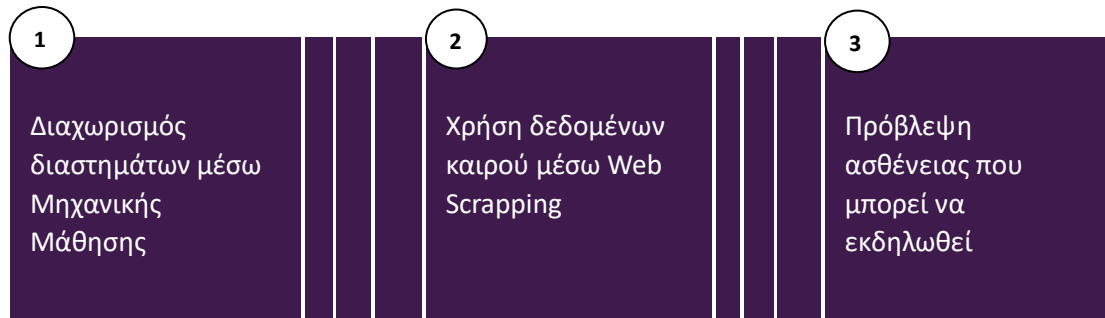
- Εύρεση των centroids:

$$c_i = \frac{1}{|N_i|} \sum x_i \quad 4-2$$

Σε σχέση με την εργασία που εκπονήθηκε, το μοντέλο είναι κατάλληλο γιατί παρέχει ένα ακριβές αποτέλεσμα και μπορεί να επεξεργαστεί δεδομένα με πολλές διαστάσεις. Στην περίπτωση της χρησιμοποιούνται 4 διαστάσεις (θερμοκρασία, υγρασία εδάφους, υγρασία αέρα, φωτεινή ακτινοβολία). Για την επιλογή του μοντέλου έχει ληφθεί υπόψη και το γεγονός ότι έχει χρησιμοποιηθεί στον τομέα της ιατρικής για την κατηγοριοποίηση των ασθενειών που παρουσίαζε το συκώτι με βάση τα συμπτώματα των ασθενών [17]. Επομένως, και στην περίπτωση του φυτού θα κατηγοριοποιηθούν οι ασθένειες του με βάση τις τιμές των παραμέτρων. Παρόλα αυτά, στο μοντέλο k-means υπάρχει ένα αβέβαιο σημείο και αυτό είναι η επιλογή του K. Δεν είναι εφικτό να είναι γνωστός ο βέλτιστος αριθμός των συστάδων. Γι' αυτό, για να παρθεί απόφαση για το K χρησιμοποιείται η τεχνική "Sillouette method" [22]. Μέσω της Sillouette method υπολογίζεται η απόδοση του k-means με βάση πολλές τιμές του K. Αυτό εφαρμόζεται με την εύρεση των ομοιοτήτων ενός σημείου κάποιας συστάδας με τα σημεία της ίδιας συστάδας και με εκείνα των υπόλοιπων συστάδων. Εάν τα σημεία μιας συστάδας έχουν μεγάλη απόσταση από τα σημεία των άλλων συστάδων τότε το αποτέλεσμα είναι το βέλτιστο. Εάν όμως είναι κοντά μεταξύ τους το αποτέλεσμα δεν οδηγεί σε ένα σωστό μοντέλο. Το εύρος των τιμών της Sillouette Method είναι από -1 έως 1. Όταν το αποτέλεσμα πλησιάζει στο 1 τότε το μοντέλο δίνει τον βέλτιστο διαχωρισμό και ο αριθμός των συστάδων είναι ο ιδανικός. Αν η τιμή της είναι 0, τότε τα σημεία βρίσκονται στο όριο ή σχετικά κοντά στο όριο μεταξύ των δύο συστάδων και αν είναι στο -1 τα σημεία βρίσκονται σε λάθος συστάδα. Επομένως, το πρόγραμμα θα είναι ικανό να βρει τον ιδανικό αριθμό συστάδων ανάλογα με το Sillouette Score. Έτσι, οι ασθένειες μπορεί να είναι πάνω από 10 ή κάποιες να συγχωνευτούν σε μία γιατί παρουσιάζουν κοινά συμπτώματα.

4.2 Πλεονεκτήματα του Plant Disease Forecasting για τους καλλιεργητές και διαδικασία υλοποίησης

Ο τομέας των καλλιεργειών με τη χρήση του Machine Learning μπορεί να καταφέρει να προβλέπει την εκδήλωση των ασθενειών μέσω μοντέλων. Αυτό αποτελεί μια πολύ μεγάλη καινοτομία, καθώς οι καλλιεργητές μπορούν να προφυλάσσονται πριν εκδηλωθεί η ασθένεια. Για να γίνει η πρόβλεψη των ασθενειών χρησιμοποιείται εκτός από την μέθοδο του K-MEANS clustering ένα σύνολο από στοιχεία που σχετίζονται με τον καιρό. Τα στοιχεία αυτά είναι η θερμοκρασία, η υγρασία και η ταχύτητα του ανέμου των επόμενων ημερών. Επιπλέον, για να γίνει πιο ακριβές το μοντέλο λαμβάνονται ανά μία ώρα. Έτσι, σε συνδυασμό με το πρόγραμμα Μηχανικής Μάθησης, το πρόγραμμα λαμβάνοντας υπόψη τα διαστήματα στα οποία εκδηλώνεται η κάθε ασθένεια, προβλέπει ποια ασθένεια είναι πιθανό να εκδηλωθεί με βάση τα δεδομένα καιρού. Ο καλλιεργητής επομένως, έχει πολύ πιο περιορισμένο ρόλο πλέον. Εφόσον μια εφαρμογή τον ειδοποιεί για την πιθανή εκδήλωση ασθενειών μπορεί και υλοποιεί ενέργειες πιο νωρίς. Επομένως, το πρόγραμμα είναι υπεύθυνο να υλοποιήσει την ακόλουθη διαδικασία:



Σχήμα 4.2: Διαδικασία δημιουργίας του μοντέλου[33]

4.2.1 Χρήση μοντέλου Μηχανικής Μάθησης

Το μοντέλο Μηχανικής Μάθησης που χρησιμοποιήθηκε περιλαμβάνει κάποια διαδικασία για να υλοποιηθεί. Περιλαμβάνει την προσθήκη δεδομένων, την κανονικοποίηση, την εύρεση του PCA, την Silhouette Method και τον διαχωρισμό των συστάδων.

Load Data

Η χρήση του μοντέλου K-MEANS CLUSTERING όπως αναφέρθηκε έχει βοηθήσει στον διαχωρισμό των διαστημάτων εκδήλωσης των ασθενειών. Για την ανάλυση του dataset που χρησιμοποιήθηκε πρέπει το πρόγραμμα να μπορεί να διαβάσει τα δεδομένα σε μορφή που να μπορεί να χρησιμοποιηθεί όπως την “csv”. Η μορφή αυτή παρουσιάζεται παρακάτω:

```
1. home_data = pd.read_csv('C:/Users/marix/Downloads/data35.csv',
2. usecols = ['Light', 'Temperature', 'Humidity' , 'Moisture' ,
3. ])
4.
5. home_data.head()
```

Σχήμα 4.3: Φόρτωση δεδομένων [34]

Scaling

Στη συνέχεια, είναι μεγάλης σημασίας να πραγματοποιηθεί το Scaling. Η χρήση του είναι αρκετά συχνή καθώς τα δεδομένα που χρησιμοποιούνται δεν έχουν παρόμοιες τιμές. Κάποιες τιμές μπορεί να είναι κοντά στο 10, κάποιες στο 600 και κάποιες στο 1000. Επομένως, πρέπει να γίνει η κανονικοποίηση των τιμών. Υλοποιείται με τη χρήση της συνάρτησης `StandardScaler`.

```
1. home_data_scaled[home_data_scaled.columns]=
2. StandardScaler().fit_transform(home_data_scaled)
3. print(home_data_scaled.describe() )
```

Σχήμα 4.4: Κανονικοποίηση [35]

Principal Component Analysis

Στη συνέχεια, οι διαστάσεις του μοντέλου πρέπει να ελαττωθούν έτσι ώστε να γίνει η αναπαράστασή με τη χρήση των καρτεσιανών συντεταγμένων (x,y). Αυτό υλοποιείται με την μέθοδο Principal Component Analysis (PCA), δηλαδή την Ανάλυση Βασικού Τμήματος. Η συγκεκριμένη, μετατρέπει τις διαστάσεις από τέσσερις σε δύο. Το script της συνάρτησης φαίνεται παρακάτω.

```
1. # Minimize the dimensions
2. pca_B = PCA(n_components = 2 )
3. pca_B_rslt = pca_B.fit_transform(df_scaled)
4. print('Explained variation per principal component:
5. {}'.format(pca_B.explained_variance_ratio_))
6. print('Cumulative variance explained by 2 principal
7. components: {:.2%}'.format(
8. np.sum(pca_B.explained_variance_ratio_)))
9. # Results from pca.components_
10. dataset_pca = pd.DataFrame(abs(pca_B.components_),
11. columns=df_scaled.columns, index=['PC_1', 'PC_2'])
12. print('\n\n', dataset_pca)
```

Σχήμα 4.5: Principal Component Analysis [36]

Silhouette Method

Στη συνέχεια, ακολουθεί η διαδικασία της εύρεσης του Silhouette Score. Το Silhouette Score είναι υπεύθυνο στο να αποφασίζει τον ιδανικό αριθμό των συστάδων με βάση τη διαδικασία που περιγράφηκε στην προηγούμενη ενότητα. Η πραγματοποίησή του γίνεται με το ακόλουθο script.

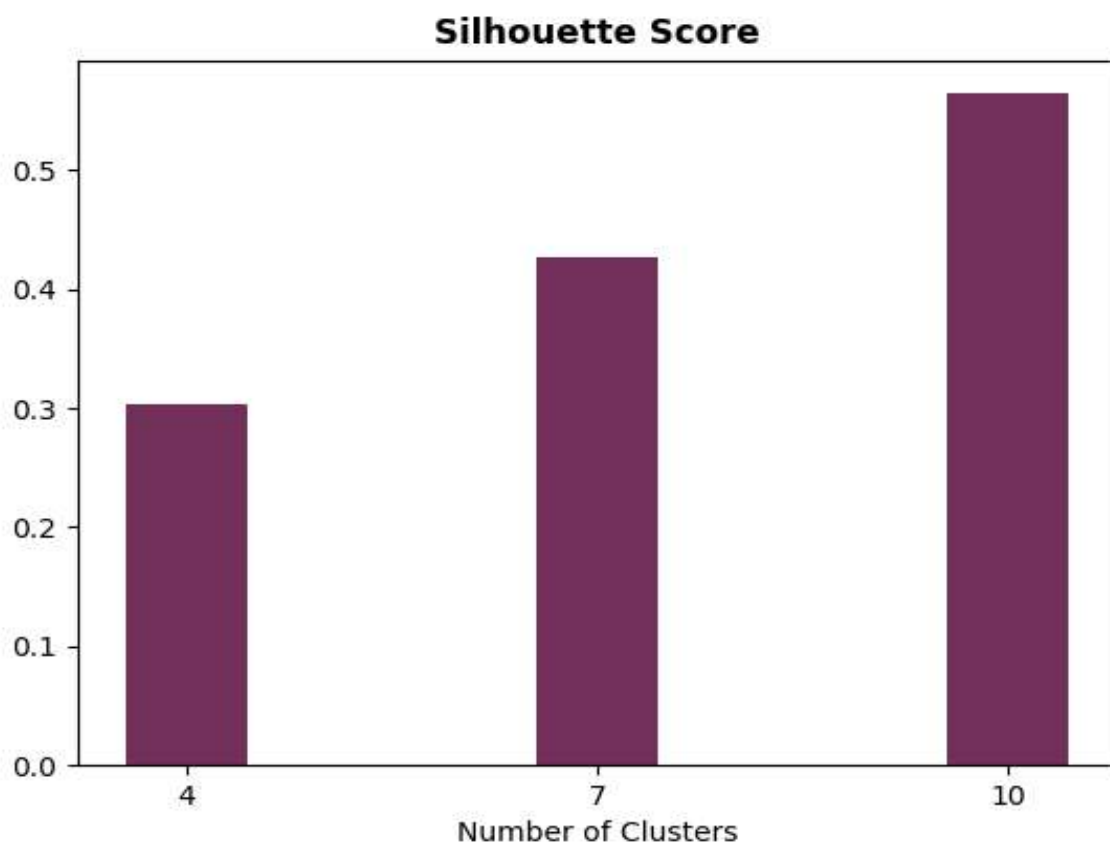
```

1. parameters = [ 4 , 7, 10]
2. # Pass number of clusters as input
3. parameter_grid = ParameterGrid({'n_clusters': parameters})
4. best_score = -1
5. kmeans_model = KMeans()      # instantiating KMeans model
6. silhouette_scores = []
7. # evaluation based on silhouette_score
8. for p in parameter_grid:
9.     kmeans_model.set_params(**p) # set current hyper parameter
10.    kmeans_model.fit(data)        # fit model on plant dataset,
11.    this will find clusters based on parameter p
12.    ss = metrics.silhouette_score(data, kmeans_model.labels_)
13.    # calculate silhouette_score
14.    silhouette_scores += [ss]      # store all the scores
15.    print('Parameter:', p, 'Score', ss)

```

Σχήμα 4.6: Silhouette Method [37]

Τα αποτελέσματα της εκτέλεσης του script είναι για τις τιμές [4, 7, 10] των συστάδων οι ακόλουθες:



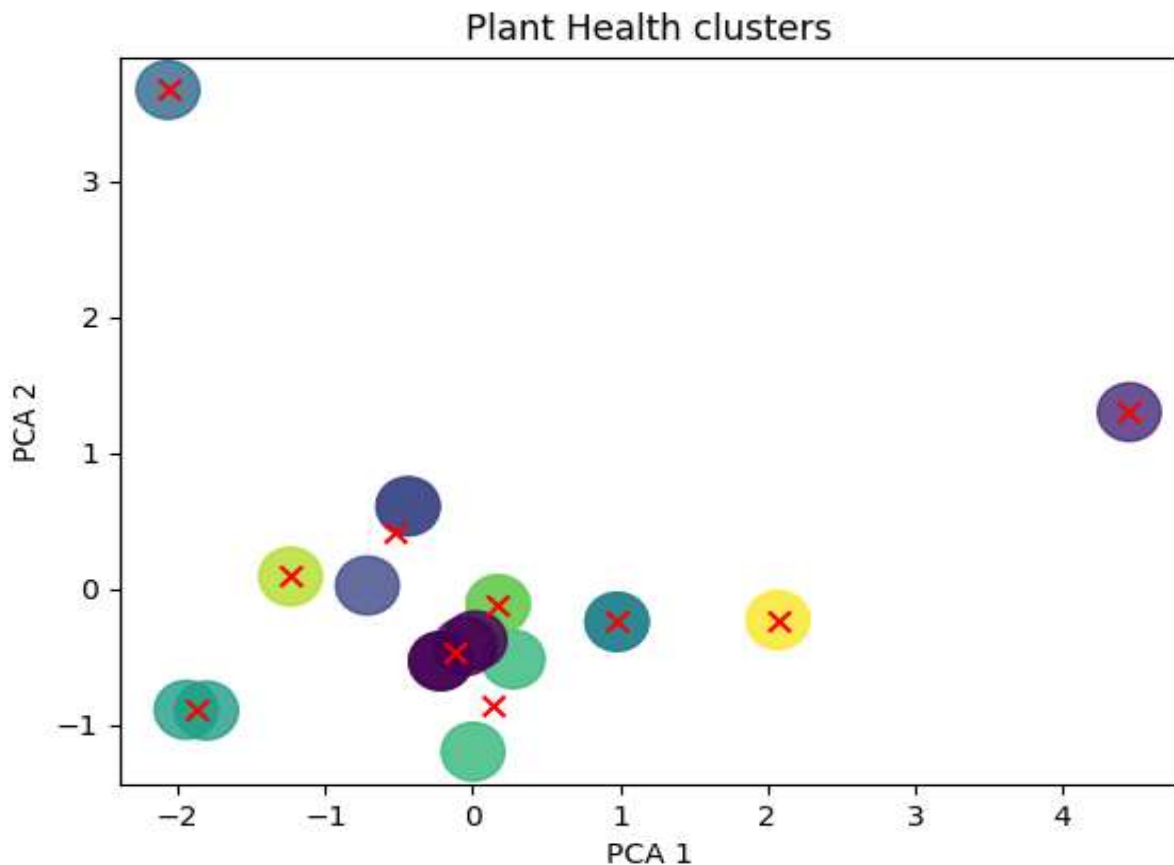
Σχήμα 4.7: Silhouette Scores για διαφορετικές τιμές των συστάδων [38]

Για την τιμή «4», το Silhouette Score είναι αρκετά χαμηλό, δηλαδή 0.3. Επομένως, ο αριθμός των συστάδων δεν είναι ο ιδανικός. Σε παρόμοια περίπτωση βρίσκεται και για «7»

συστάδες αφού το Silhouette score είναι 0.4. Τέλος, για τον αριθμό 10 το Silhouette score είναι 0.6 το οποίο είναι το βέλτιστο. Επομένως, θα γίνει διαχωρισμός με δέκα συστάδες, δηλαδή όσες είναι και οι ασθένειες του dataset.

Διαχωρισμός

Τα αποτελέσματα του μοντέλου γίνονται πιο εύληπτα στο επόμενο σχήμα:



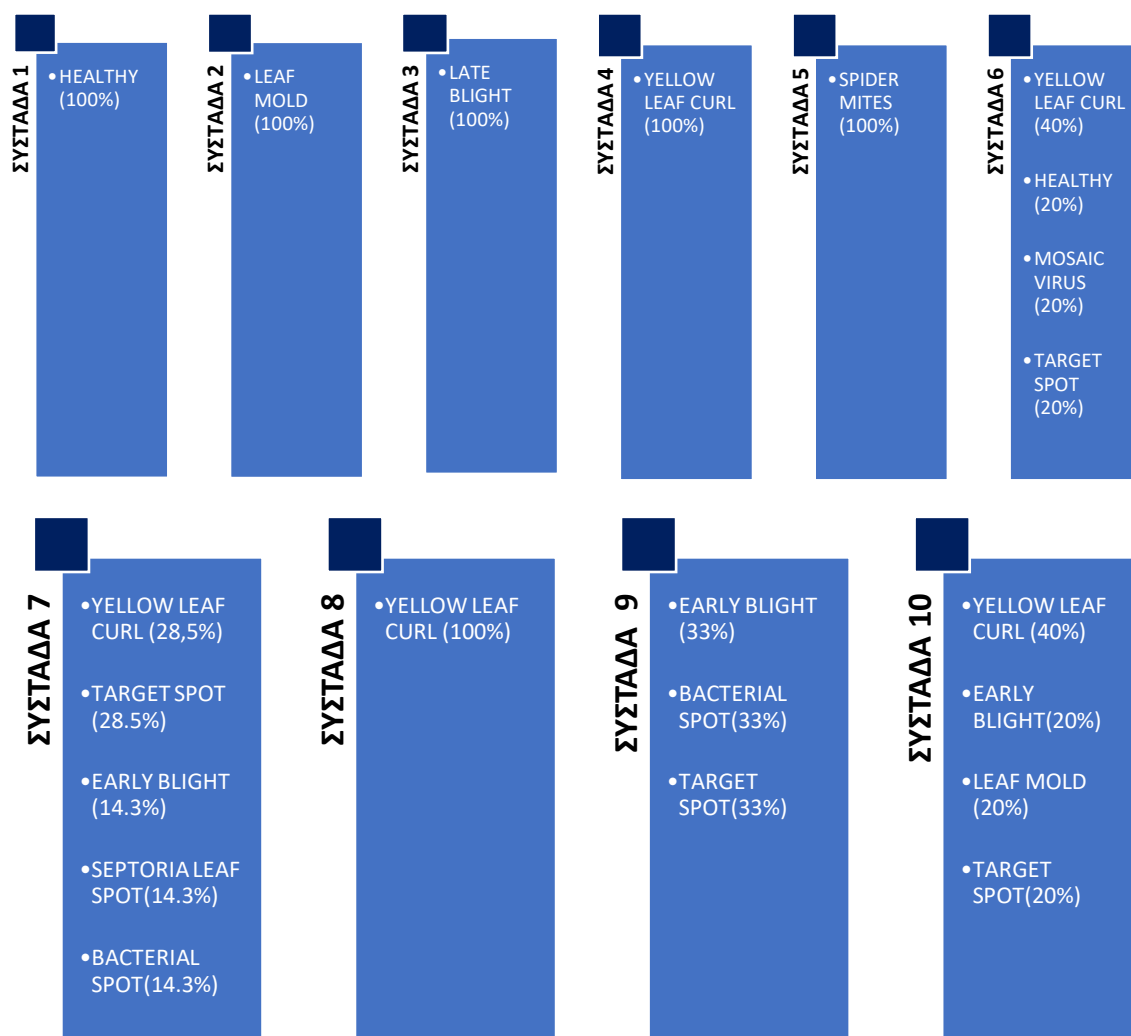
Σχήμα 4.8: Αποτελέσματα του μοντέλου [39]

Η διαδικασία της λήψης του dataset ήταν περίπου 3 μήνες, ανά 4 – 5 ημέρες. Το μοντέλο χρησιμοποίησε ως βασικό στοιχείο διαχωρισμού των συστάδων το humidity με τιμή 0.56 (importance). Ως επόμενο χρησιμοποίησε τη θερμοκρασία με τιμή 0.55 (importance). Ως τρίτο βασικό στοιχείο χρησιμοποίησε την υγρασία εδάφους με τιμή 0.54 (importance). Η φωτεινή ακτινοβολία δεν αποτέλεσε τόσο βασικό στοιχείο διότι ήταν σχεδόν αμετάβλητη.

Με βάση το μοντέλο οι συστάδες που δημιουργήθηκαν περιέχουν τα ακόλουθα δεδομένα:

-

4.2.2 Αποτελέσματα



Σχήμα 4.9: Συστάδες 1 – 10 [40]

Γενικά, το μοντέλο παρουσιάζει ευνοϊκά αποτελέσματα. Κάποιες συστάδες έχουν 100% επιτυχία, καθώς περιλαμβάνουν μία ασθένεια. Έτσι, είναι πιο εύκολο να γίνει κατανοητό το διάστημα εκδήλωσης της ασθένειας. Σε άλλες συστάδες περιέχονται παραπάνω από μία ασθένειες το οποίο καθιστά πιο δύσκολη την ανάλυση. Η ασθένεια Yellow Leaf Curl, για παράδειγμα εμφανίζεται σε σχεδόν όλες τις συστάδες. Είναι η πιο συχνή ασθένεια και μπορεί να εκδηλωθεί σε πολλές συνθήκες. Επιπρόσθετα, η ασθένεια “Early Blight” εμφανίζεται μαζί με την ασθένεια “Target Spot”. Αυτό οφείλεται στο γεγονός ότι έχουν πολλά κοινά στοιχεία και μπορούν να θεωρηθούν ως ίδια ασθένεια. Η ασθένεια “Bacterial Spot” εμφανίζεται σε μεγάλο βαθμό μαζί με τις προηγούμενες. Στη συνέχεια αναλύονται τα διαστήματα εμφάνισης της κάθε ασθένειας με βάση τη συστάδα στην οποία βρίσκονται.

ΣΥΣΤΑΔΑ 1	ΣΥΣΤΑΔΑ 2	ΣΥΣΤΑΔΑ 5
ΦΩΤΕΙΝΗ ΑΚΤΙΝΟΒΟΛΙΑ [207] ΘΕΡΜΟΚΡΑΣΙΑ [34] ΥΓΡΑΣΙΑ ΑΕΡΑ [30] ΥΓΡΑΣΙΑ ΕΔΑΦΟΥΣ [209]	ΦΩΤΕΙΝΗ ΑΚΤΙΝΟΒΟΛΙΑ [538,699] ΘΕΡΜΟΚΡΑΣΙΑ [26,27] ΥΓΡΑΣΙΑ ΑΕΡΑ [52,60] ΥΓΡΑΣΙΑ ΕΔΑΦΟΥΣ [1817,1980]	ΦΩΤΕΙΝΗ ΑΚΤΙΝΟΒΟΛΙΑ [698] ΘΕΡΜΟΚΡΑΣΙΑ [34] ΥΓΡΑΣΙΑ ΑΕΡΑ [35] ΥΓΡΑΣΙΑ ΕΔΑΦΟΥΣ [1942]
ΣΥΣΤΑΔΑ 3	ΣΥΣΤΑΔΑ 4	
ΦΩΤΕΙΝΗ ΑΚΤΙΝΟΒΟΛΙΑ[601,699] ΘΕΡΜΟΚΡΑΣΙΑ [19,26] ΥΓΡΑΣΙΑ ΑΕΡΑ [58,63] ΥΓΡΑΣΙΑ ΕΔΑΦΟΥΣ [1980,2005]	ΦΩΤΕΙΝΗ ΑΚΤΙΝΟΒΟΛΙΑ [600,698] ΘΕΡΜΟΚΡΑΣΙΑ [21,32] ΥΓΡΑΣΙΑ ΑΕΡΑ [52,75] ΥΓΡΑΣΙΑ ΕΔΑΦΟΥΣ [1875,2293]	

Σχήμα 4.10: Συστάδες 1 – 5 [41]

Για τις συστάδες που προηγήθηκαν, η ανάλυση είναι πολύ πιο εύκολη. Ο λόγος είναι ότι περιέχουν μια ασθένεια. Στην πρώτη συστάδα υπάρχει η περίπτωση στην οποία το φυτό ήταν υγιές. Το φυτό ήταν υγιές σε πολύ λίγες περιπτώσεις. Στη συγκεκριμένη, η φωτεινή ακτινοβολία είναι σε χαμηλά επίπεδα σε σχέση με τις επόμενες περιπτώσεις, η θερμοκρασία είναι σε υψηλά επίπεδα, και η υγρασία σε πολύ χαμηλά. Επομένως, το φυτό χαρακτηρίζεται από έλλειψη νερού. Η καλλιέργεια της ντομάτας χρειάζεται ένα πολύ θερμό και υγρό περιβάλλον για την βέλτιστη υγεία. Στη συγκεκριμένη περίπτωση, το υγρό περιβάλλον δεν υπάρχει. Παρόλα αυτά, επειδή ήταν η πρώτη μέτρηση της ντομάτας, δεν έχει προλάβει να αναπτύξει κάποια ασθένεια. Στην δεύτερη συστάδα αναπτύσσεται η ασθένεια Leaf Mold. Περιλαμβάνει μέτρια προς υψηλή θερμοκρασία, υψηλή υγρασία και υψηλή ακτινοβολία. Η συγκεκριμένη συστάδα θα μπορούσε να θεωρηθεί ότι παρέχει αισιόδοξο αποτέλεσμα καθώς η ασθένεια αυτή όπως αναφέρθηκε στο προηγούμενο κεφάλαιο χαρακτηρίζεται από υψηλή υγρασία. Η συστάδα 3 περιέχει το late blight το οποίο χαρακτηρίζεται από μέτρια θερμοκρασία, και υψηλή υγρασία. Το αποτέλεσμα είναι επίσης αισιόδοξο καθώς στο προηγούμενο κεφάλαιο η συγκεκριμένη ασθένεια εμφανίζεται υπό αυτά τα στοιχεία. Στην συστάδα 4 περιέχεται η ασθένεια Yellow Leaf Curl. Η συγκεκριμένη ασθένεια ευδοκimeί σε περιπτώσεις που υπάρχει υψηλή θερμοκρασία. Επομένως, θα εμφανιστεί και στα επόμενες συστάδες καθώς σε σχεδόν όλες τις περιπτώσεις η θερμοκρασία είναι υψηλή. Στη συστάδα 5 περιέχεται η ασθένεια Spider Mites. Η συγκεκριμένη εμφανίζεται σε περιπτώσεις που το περιβάλλον είναι θερμό και υπάρχει απουσία νερού. Στη συγκεκριμένη περίπτωση, τα στοιχεία έχουν αρκετά μεγάλη σχέση με τις συνθήκες ανάπτυξης.

ΣΥΣΤΑΔΑ 6	ΣΥΣΤΑΔΑ 7
ΦΩΤΕΙΝΗ ΑΚΤΙΝΟΒΟΛΙΑ [600,698] ΘΕΡΜΟΚΡΑΣΙΑ [21,32] ΥΓΡΑΣΙΑ ΑΕΡΑ [52,75] ΥΓΡΑΣΙΑ ΕΔΑΦΟΥΣ [1875,2293]	ΦΩΤΕΙΝΗ ΑΚΤΙΝΟΒΟΛΙΑ [600,698] ΘΕΡΜΟΚΡΑΣΙΑ [21,32] ΥΓΡΑΣΙΑ ΑΕΡΑ [52,75] ΥΓΡΑΣΙΑ ΕΔΑΦΟΥΣ [1875,2293]
ΦΩΤΕΙΝΗ ΑΚΤΙΝΟΒΟΛΙΑ [698] ΘΕΡΜΟΚΡΑΣΙΑ [33,34] ΥΓΡΑΣΙΑ ΑΕΡΑ [70] ΥΓΡΑΣΙΑ ΕΔΑΦΟΥΣ [2018]	ΦΩΤΕΙΝΗ ΑΚΤΙΝΟΒΟΛΙΑ [634,699] ΘΕΡΜΟΚΡΑΣΙΑ [23,32] ΥΓΡΑΣΙΑ ΑΕΡΑ [58,75] ΥΓΡΑΣΙΑ ΕΔΑΦΟΥΣ [2134,2304]
ΦΩΤΕΙΝΗ ΑΚΤΙΝΟΒΟΛΙΑ [634] ΘΕΡΜΟΚΡΑΣΙΑ [32] ΥΓΡΑΣΙΑ ΑΕΡΑ [58] ΥΓΡΑΣΙΑ ΕΔΑΦΟΥΣ [2134]	ΦΩΤΕΙΝΗ ΑΚΤΙΝΟΒΟΛΙΑ [690,699] ΘΕΡΜΟΚΡΑΣΙΑ [24,30] ΥΓΡΑΣΙΑ ΑΕΡΑ [48,75] ΥΓΡΑΣΙΑ ΕΔΑΦΟΥΣ [1839,2294]
ΦΩΤΕΙΝΗ ΑΚΤΙΝΟΒΟΛΙΑ [634,699] ΘΕΡΜΟΚΡΑΣΙΑ [23,32] ΥΓΡΑΣΙΑ ΑΕΡΑ [58,75] ΥΓΡΑΣΙΑ ΕΔΑΦΟΥΣ [2134,2304]	ΦΩΤΕΙΝΗ ΑΚΤΙΝΟΒΟΛΙΑ [699] ΘΕΡΜΟΚΡΑΣΙΑ [23] ΥΓΡΑΣΙΑ ΑΕΡΑ [75] ΥΓΡΑΣΙΑ ΕΔΑΦΟΥΣ [2304]
	ΦΩΤΕΙΝΗ ΑΚΤΙΝΟΒΟΛΙΑ [690,699] ΘΕΡΜΟΚΡΑΣΙΑ [24,30] ΥΓΡΑΣΙΑ ΑΕΡΑ [48,75] ΥΓΡΑΣΙΑ ΕΔΑΦΟΥΣ [1839,2292]

Σχήμα 4.11: Συστάδες 6, 7 [42]

Στις συστάδες που προηγήθηκαν, η ανάλυση είναι πιο δύσκολη επειδή περιέχουν αρκετές ασθένειες. Στη συστάδα 6 περιλαμβάνεται η Yellow Leaf Curl, Healthy, Mosaic Virus και Target Spot. Το Target spot εμφανίζεται σε περιπτώσεις που υπάρχει πολύ μεγάλη υγρασία, κάτι που περιέχει η συγκεκριμένη συστάδα. Η Yellow Leaf Curl εμφανίζεται σε υψηλές θερμοκρασίες, στοιχείο που επίσης περιέχει η συστάδα. Το Mosaic Virus δεν σχετίζεται με τις συνθήκες αυτές αλλά βασική του αιτία είναι κάποιες ρωγμές που πιθανόν να έχει το φυτό. Πιθανό είναι η συστάδα να περιέχει αυτές τις ασθένειες γιατί χαρακτηρίζονται από υψηλές θερμοκρασίες και υψηλή υγρασία. Στη συστάδα 7 περιέχεται η Yellow Leaf Curl, η Early Blight, Target Spot, Septoria Leaf Spot και Bacterial Spot. Η πρώτη

εμφανίστηκε για τους ίδιους λόγους που αναφέρθηκαν πριν. Η Early Blight και Target Spot είναι λογικό να εμφανιστούν μαζί, καθώς αποτελούν σχεδόν παρόμοια ασθένεια. Οι διαφορές που εμφανίζουν είναι ελάχιστες και οφείλονται στο μέγεθος των κηλίδων που περιέχουν. Η Septoria Leaf Spot και το Bacterial Spots οφείλονται στην παρουσία υψηλής υγρασίας, στοιχείο που επιβεβαιώνεται από τα στοιχεία της συστάδας.

ΣΥΣΤΑΔΑ 8	ΣΥΣΤΑΔΑ 9	ΣΥΣΤΑΔΑ 10
ΦΩΤΕΙΝΗ ΑΚΤΙΝΟΒΟΛΙΑ [600,698] ΘΕΡΜΟΚΡΑΣΙΑ [21,32] ΥΓΡΑΣΙΑ ΑΕΡΑ [52,75] ΥΓΡΑΣΙΑ ΕΔΑΦΟΥΣ [1875,2293]	ΦΩΤΕΙΝΗ ΑΚΤΙΝΟΒΟΛΙΑ [690,699] ΘΕΡΜΟΚΡΑΣΙΑ [24,30] ΥΓΡΑΣΙΑ ΑΕΡΑ [48,75] ΥΓΡΑΣΙΑ ΕΔΑΦΟΥΣ [1839,2294]	ΦΩΤΕΙΝΗ ΑΚΤΙΝΟΒΟΛΙΑ [600,698] ΘΕΡΜΟΚΡΑΣΙΑ [21,32] ΥΓΡΑΣΙΑ ΑΕΡΑ [52,75] ΥΓΡΑΣΙΑ ΕΔΑΦΟΥΣ [1875,2293]
	ΦΩΤΕΙΝΗ ΑΚΤΙΝΟΒΟΛΙΑ [690,699] ΘΕΡΜΟΚΡΑΣΙΑ [24,30] ΥΓΡΑΣΙΑ ΑΕΡΑ [48,75] ΥΓΡΑΣΙΑ ΕΔΑΦΟΥΣ [1839,2292]	ΦΩΤΕΙΝΗ ΑΚΤΙΝΟΒΟΛΙΑ [690,699] ΘΕΡΜΟΚΡΑΣΙΑ [24,30] ΥΓΡΑΣΙΑ ΑΕΡΑ [48,75] ΥΓΡΑΣΙΑ ΕΔΑΦΟΥΣ [1839,2294]
	ΦΩΤΕΙΝΗ ΑΚΤΙΝΟΒΟΛΙΑ [634,699] ΘΕΡΜΟΚΡΑΣΙΑ [23,32] ΥΓΡΑΣΙΑ ΑΕΡΑ [58,75] ΥΓΡΑΣΙΑ ΕΔΑΦΟΥΣ [2134,2304]	ΦΩΤΕΙΝΗ ΑΚΤΙΝΟΒΟΛΙΑ [538,699] ΘΕΡΜΟΚΡΑΣΙΑ [26,27] ΥΓΡΑΣΙΑ ΑΕΡΑ [52,60] ΥΓΡΑΣΙΑ ΕΔΑΦΟΥΣ [1817,1980]
		ΦΩΤΕΙΝΗ ΑΚΤΙΝΟΒΟΛΙΑ [634,699] ΘΕΡΜΟΚΡΑΣΙΑ [23,32] ΥΓΡΑΣΙΑ ΑΕΡΑ [58,75] ΥΓΡΑΣΙΑ ΕΔΑΦΟΥΣ [2134,2304]

Σχήμα 4.12: Συστάδες 8 – 10 [43]

Στη συστάδα 8 περιέχεται η Yellow Leaf Curl και στο 9 οι Early Blight, Bacterial Spot και Target Spot. Οι συγκεκριμένες ασθένειες σε πολλές περιπτώσεις εμφανίζονταν μαζί. Αυτό σημαίνει ότι έχουν παρόμοιες αιτίες δημιουργίας. Στη συστάδα 10 περιέχονται οι Yellow Leaf Curl, Early Blight, Leaf Mold και Target Spot. Ο συνδυασμός της Early Blight και Target Spot είναι για τον λόγο που αναφέρθηκε προηγουμένως. Η παρουσία του Leaf Mold πιθανό να έχει σχέση με την υψηλή υγρασία από την οποία χαρακτηρίζεται η συστάδα που την περιέχει.

Οι βασικοί παράγοντες που επηρεάζουν επομένως την κατάσταση του φυτού είναι η υγρασία και η θερμοκρασία. Η υψηλή θερμοκρασία δηλαδή μεγαλύτερη από 21 οδηγεί σε πολλές περιπτώσεις στην ασθένεια Yellow Leaf Curl. Η υψηλή υγρασία μεγαλύτερη του 52% μπορεί να προκαλέσει τις ασθένειες Leaf Mold, Early Blight, Target Spot. Οι ασθένειες αυτές εμφανίζονται αρκετά μαζί καθώς χρειάζονται παρόμοια ποσοστά υγρασίας. Η ασθένεια

bacterial spots εμφανίζεται επίσης σε παρόμοια ποσοστά υγρασίας αλλά πιθανό να περιλαμβάνει ένα μεγαλύτερο διάστημα από τις προηγούμενες. Η late blight περιλαμβάνει λίγο χαμηλότερη υγρασία και πιο χαμηλές θερμοκρασίες. Η Septoria Leaf Spot περιλαμβάνει υψηλή υγρασία αλλά χαμηλότερη θερμοκρασία. Ο διαχωρισμός επομένως έγινε σε μεγάλο βαθμό με βάση την υγρασία και τις διαφοροποιήσεις που εμφανίζει.

4.2.2 Δεδομένα καιρού

Για την εκπόνηση της εργασίας και την υλοποίηση του Web Scrapping έχει γίνει η προσθήκη των δεδομένων καιρού από την ιστοσελίδα www.meteo.gr. Η ιστοσελίδα αυτή περιέχει δεδομένα που σχετίζονται με τις παραμέτρους που αναφέρθηκαν. Ο κώδικας που χρησιμοποιήθηκε για την υλοποίηση του Web Scrapping περιλαμβάνει κάποια χαρακτηριστικά. Η μεταβλητή “url” παίρνει το url της σελίδας. Η μεταβλητή page είναι υπεύθυνη στο να στείλει το αίτημα στη σελίδα Meteo για χρήση των δεδομένων της. Η μεταβλητή soup είναι υπεύθυνη στο να πάρει την html σελίδα της Meteo έτσι ώστε να χρησιμοποιηθούν τα δεδομένα του προγράμματος. Παρακάτω, γίνεται πιο εύληπτος ο κώδικας και τα αποτελέσματα των μεταβλητών.

```
1. url = https://www.meteo.gr/cf.cfm?city_id=1
2. page = requests.get(url)
3. print(page)
4. soup = BeautifulSoup(page.content, 'html.parser')
5. print(soup)
```

Σχήμα 4.13: Μεταβλητές για την χρησιμοποίηση του url [44]

```
meta content="Προγνώσεις καιρού για ΘΕΣΣΑΛΟΝΙΚΗ νομού ΘΕΣΣΑΛΟΝΙΚΗΣ
(MAKEΔΟΝΙΑ) - THESSALONIKI, MACEDONIA, Greece" name="title"> <meta
content="Προγνώσεις καιρού για ΘΕΣΣΑΛΟΝΙΚΗ." name="description">
```

Σχήμα 4.14 : Τμήμα των αποτελεσμάτων των μεταβλητών page και soup [45]

Στη συνέχεια, από το html script εμφανίζονται τα δεδομένα που σχετίζονται με τον καιρό. Από το “soup” η συνάρτηση βρίσκει τα δεδομένα που απαιτεί το πρόγραμμα. Με την τιμή y_data οι τιμές παίρνουν μορφή πίνακα.

```
1. table = soup.find_all("table")
2. y_data = [row.text.splitlines() for row in table]
3. y_data = y_data[:-8]
4. for i in range(len(y_data)):
5.     y_data[i] = y_data[i][2:len(y_data[i]):1]
6. print(y_data)
```

Σχήμα 4.15: Διαχωρισμός κειμένου [46]

Το αποτέλεσμα έχει την ακόλουθη μορφή:

```
y= [['', ''], [' Τεταρτη 27ΣΕΠΤΕΜΒΡΙΟΥ ', ' Ανατολή: 07:19\χα0 - Δύση 19:17 ', '', '', '', '',
'18:00', '', '', '25°C',
'48% υγρ.', '', '', '48% υγρ.', '', '', '', '', '2 Μπφ ΝΔ', '9 Km/h', '', '', '', '',
'', 'ΑΡΚΕΤΑ ΣΥΝΝΕΦΑ ', '', '', '', 'ΑΡΚΕΤΑ ΣΥΝΝΕΦΑ ', '', '', '', '21:00',
'', '22°C', '50% υγρ.', '', '50% υγρ.', '', '2 Μπφ Β', '9 Km/h',
'', 'ΑΡΚΕΤΑ ΣΥΝΝΕΦΑ ', 'ΑΡΚΕΤΑ ΣΥΝΝΕΦΑ ',
'', '21:00', '22°C', '50% υγρ.', '50% υγρ.', '2 Μπφ Β', '9 Km/h',
'', 'ΑΡΚΕΤΑ ΣΥΝΝΕΦΑ ', 'ΑΡΚΕΤΑ ΣΥΝΝΕΦΑ ',
'', '21:00', '22°C', '50% υγρ.', '50% υγρ.', '2 Μπφ Β', '9 Km/h',
'', 'ΑΡΚΕΤΑ ΣΥΝΝΕΦΑ ', 'ΑΡΚΕΤΑ ΣΥΝΝΕΦΑ ']
```

Σχήμα 4.16: Τμήμα των αποτελεσμάτων σε μορφή κειμένου [47]

Ο πίνακας αυτός δεν θα μπορούσε να θεωρηθεί αρκετά διαχειρίσιμος καθώς αποτελείται από αλφαριθμητικά και από αρκετά κενά. Επομένως, πρέπει τα κενά να αφαιρεθούν και οι παράμετροι «θερμοκρασία», «υγρασία» και «ταχύτητα του αέρα» να αποτελέσουν πίνακες και να είναι floats. Οι επόμενες μεταβλητές μετατρέπουν τον πίνακα που προηγήθηκε σε dataframe (pd.DataFrame), αφαιρούν τις θέσεις που δεν έχουν κανένα στοιχείο (apply(pd.to_numeric, errors = 'coerce')) και αφαιρούν τα κενά (df.dropna).

```
1. for i in range(2): v = {'text':
2. ['.'.'join(re.findall("\d+",str(y_data[i][j].split()[:3]))))for j
3. in range(len(y_data[i]))] }
4. df = pd.DataFrame(v )
5. df = df.apply(pd.to_numeric, errors = 'coerce')
6. df = df.dropna()
7. print(df)
```

Σχήμα 4.17: Διαχωρισμός των χρήσιμων δεδομένων [48]

Έτσι το αποτέλεσμα είναι το ακόλουθο:

	text
0	27.00
1	7.19
7	18.00
...	...

Σχήμα 4.18: Δημιουργία πίνακα με τις παραμέτρους [49]

Έχει δημιουργηθεί ένα dataframe το οποίο έχει μεγάλη δυσκολία στη διαχείριση. Έτσι, είναι πρωτεύουσας σημασίας να μειωθεί το μέγεθός του και να δημιουργηθούν πίνακες που περιέχουν τις παραμέτρους. Αυτό επιτυγχάνεται με παρατήρηση. Η υγρασία τυπώνεται δύο φορές από το πρόγραμμα. Αυτό μπορεί να χρησιμοποιηθεί στην εμφάνισή της. Με παρόμοια λογική βρίσκονται και οι υπόλοιπες παράμετροι. Πιο συγκεκριμένα, η θερμοκρασία βρίσκεται δύο θέσεις πριν την υγρασία, ενώ η ταχύτητά του δύο θέσεις μετά. Σε άλλες περιπτώσεις, μπορεί να υπάρχουν άλλου είδους συνδυασμοί που να βοηθούν στη δημιουργία πινάκων. Παρακάτω, γίνεται κατανοητό το documentation για τη δημιουργία πινάκων

```
1.[for i in range (len(humidity1) ):  
2.    if humidity1[i-1] == humidity1[i]:  
3.        b[i] = humidity1[i]  
4.        value = float(''.join(map(str,humidity1[i]))  
5.        b[i] = value  
6.    print(b[i])
```

Σχήμα 4.19: Διαχωρισμός των στοιχείων [50]

Επιπλέον, η τιμή value είναι υπεύθυνη στο να μετατρέπει τις τιμές των πινάκων σε floats με στόχο να μπορεί να υπολογιστεί ο μέσος όρος των τιμών των παραμέτρων υγρασίας, θερμοκρασίας και ταχύτητας για 3 ημέρες. Το αποτέλεσμα είναι το ακόλουθο για τη θερμοκρασία:

```
25.0 22.0 21.0 20.0 20.0 20.0 25.0 26.0 24.0  
  
22.0 21.0 21.0 20.0 21.0 26.0 28.0 26.0 23.0  
  
the aver is 23.11111111111111
```

Σχήμα 4.20: Μετατροπή σε float και υπολογισμός του μέσου [51]

4.2.3 Πρόβλεψη πιθανών ασθενειών

Από τη δημιουργία διαστημάτων και από τον μέσο όρο των δεδομένων καιρού για τις 3 επόμενες ημέρες το πρόγραμμα είναι ικανό να προβλέψει ποια ασθένεια θα εκδηλωθεί. Πιο συγκεκριμένα μετά την υλοποίηση του προγράμματος εμφανίζονται τρεις τιμές: ο μέσος όρος της υγρασίας, της θερμοκρασίας και της ταχύτητας του ανέμου. Οι μέσοι όροι στη συνέχεια εισέρχονται στα διαστήματα εκδήλωσης των ασθενειών και σε εκείνα που ταιριάζουν, θα εμφανίσουν και τη συγκεκριμένη ασθένεια. Η ταχύτητα του ανέμου, μπορεί να προκαλέσει την ασθένεια Mosaic Virus καθώς οφείλεται σε ρωγμές του φυτού οι οποίες μπορεί να δημιουργηθούν από μεγάλη δύναμη που ασκεί ο άνεμος. Παρακάτω εμφανίζεται τμήμα από τα διαστήματα εκδήλωσης των ασθενειών.

```

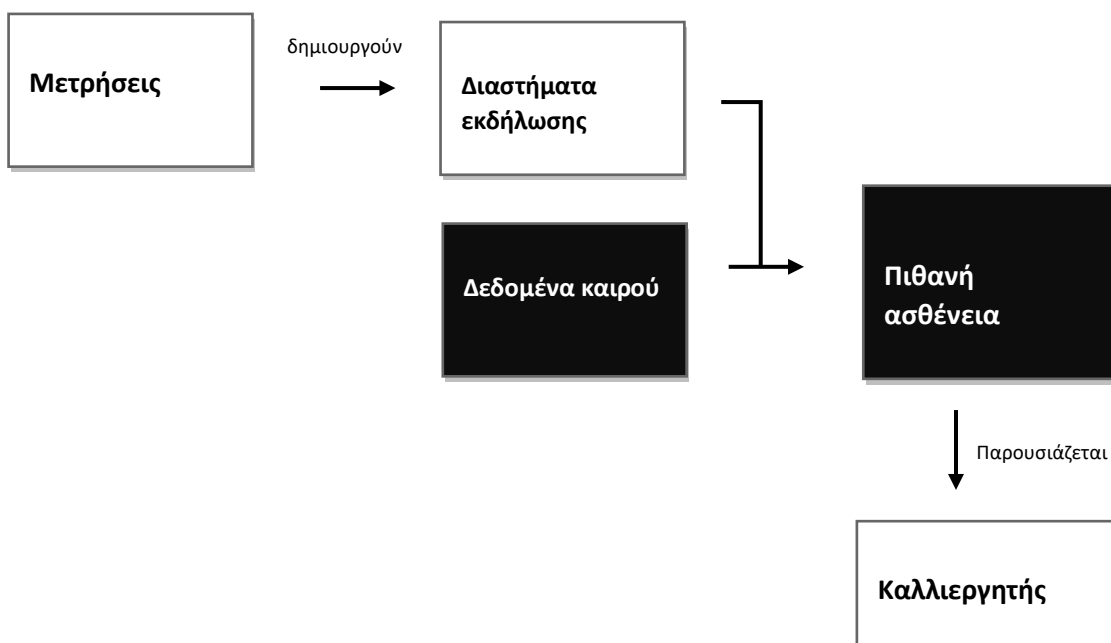
1. #HEALTHY
2. if d97 > 65.0 and d97 < 75.0 and d98 > 33.0 and d98 < 34.0:
3.   print("The plant is going to be healthy for the next 3 days")
4. #LEAF MOLD
5. if d97 > 52.0 and d97 < 60.0 and d98 > 26.0 and d98 < 27.0:
6.   print("The plant might have leaf mold in the next 3 days")

```

Σχήμα 4.21: Διαστήματα της υγείας του[52]

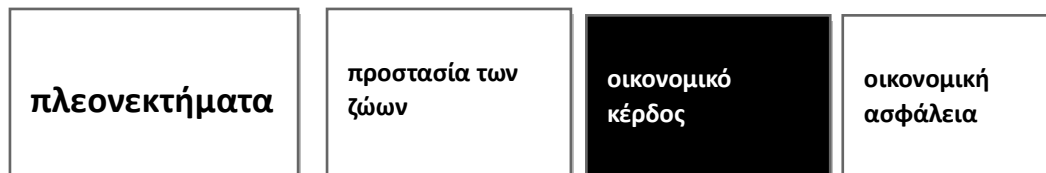
Το αποτέλεσμα είναι η τύπωση του «The plant is going to have leaf mold in the next 3 days». Έτσι ο καλλιεργητής γνωρίζει από νωρίς και μπορεί να υλοποιήσει τις κατάλληλες διαδικασίες για να προστατέψει την καλλιέργειά του.

Η τύπωση του μηνύματος της επερχόμενης ασθένειας είναι ικανή στο να βοηθήσει τον καλλιεργητή να κατανοήσει το τι χρειάζεται να κάνει για να μην εκδηλωθεί. Πιο συγκεκριμένα, η ασθένεια Yellow Leaf Curl έχει βασική αιτία δημιουργίας την έκθεση σε υψηλές θερμοκρασίες. Αν η πρόβλεψη του καιρού περιέχει θερμοκρασίες μεγαλύτερες από 21 βαθμούς τότε ο καλλιεργητής μπορεί να τοποθετήσει κάποια επιφάνεια που να μετατρέπει το περιβάλλον σε πιο σκιερό από τις προηγούμενες ημέρες. Έτσι λοιπόν, η καλλιέργεια δεν θα περιέχει αυτή την ασθένεια. Έτσι, έχουν δημιουργηθεί κάποια διαστήματα εκδήλωσης τα οποία χρησιμοποιούνται για την διαπίστωση αυτή. Το πρόγραμμα Μηχανικής Μάθησης χρησιμοποιήθηκε στο να κρίνει με μαθηματικό τρόπο το πως διαχωρίζονται τα διαστήματα εκδήλωσης καθώς και να εντοπίσει το ποιες περιπτώσεις έχουν κοινά έτσι ώστε να τις τοποθετήσει στην ίδια συστάδα. Το πρόγραμμα Μηχανικής Μάθησης δείχνει την προσέγγιση του διαχωρισμού των ασθενειών με βάση ένα μοντέλο. Έτσι δημιουργεί συστάδες που περιέχουν ένα ή πολλά στοιχεία. Στο επόμενο σχήμα γίνεται πιο εύληπτη η παρουσίαση του αποτελέσματος.



Σχήμα 4.22: Παρουσίαση του αποτελέσματος στον καλλιεργητή [53]

Με βάση τα αποτελέσματα ο καλλιεργητής θα είναι σε θέση να μπορεί σώσει την καλλιέργειά του. Έτσι, θα βελτιωθεί η οικονομία του, καθώς θα αυξηθεί η σοδειά του. Επιπλέον, δεν θα χρειάζεται να προσφεύγει στη χρήση πιθανών τοξικών ουσιών για την αντιμετώπιση των προβλημάτων που έχουν αναπτυχθεί. Οι τοξικές ουσίες είναι επιβλαβείς για μεγάλο πλήθος ζώων αλλά και φυτών και γι' αυτό δεν πρέπει να γίνεται συχνή χρήση τους. Επιπλέον, η οικονομία του θα είναι ασφαλής. Με την βελτιστοποίηση της υγείας της καλλιέργειάς του πριν την εκδήλωση ασθενειών θα μπορεί να χρησιμοποιεί πιθανόν κατασκευές για την προστασία της.



Σχήμα 4.23: Πλεονεκτήματα [54]

Κεφάλαιο 5

Αποτελέσματα και εργασίες

5.1 Αποτελέσματα

Με βάση τις μετρήσεις που έχουν υλοποιηθεί, τα στοιχεία του προγράμματος Μηχανικής Μάθησης και της χρήσης δεδομένων καιρού έχουν βγει κάποια αποτελέσματα. Οι μετρήσεις είχαν αρκετό ενδιαφέρον καθώς παρουσίαζαν την λειτουργία των IoT συσκευών, καθώς και την επικοινωνία τους με τον μικροεπεξεργαστή. Οι μετρήσεις υλοποιούνταν κάθε 4-5 ημέρες για δύο με τρεις μήνες. Αυτό μπορεί να αποτελεί περιορισμό στην ακρίβεια των αποτελεσμάτων καθώς το dataset των παραμέτρων της θερμοκρασίας, της υγρασίας του αέρα, της υγρασίας του εδάφους και της φωτεινής ακτινοβολίας είναι πιο περιορισμένο. Πιθανό αν οι μετρήσεις γίνονταν σε μεγαλύτερο χρονικό διάστημα να υπήρχε ακόμη μεγαλύτερη ακρίβεια των αποτελεσμάτων. Επιπλέον, το μοντέλο K-MEANS clustering ήταν πολύ ικανοποιητικό και επιλέχθηκε βάσει άλλων μοντέλων που έχουν υλοποιηθεί στον τομέα των καλλιεργειών. Το εμπόδιο που μπορεί να είχε το συγκεκριμένο μοντέλο ήταν η χρήση του βέλτιστου αριθμού των συστάδων. Επιπλέον, η ανάλυση των αποτελεσμάτων για τον διαχωρισμό των συστάδων είχε δυσκολίες. Αυτό οφειλόταν στο γεγονός ότι κάποιες συστάδες δεν είχαν πλήρως μία ασθένεια αλλά περίπου τρεις. Για την περίπτωση των δεδομένων καιρού το πρόγραμμα ήταν απόλυτα ακριβές. Η δυσκολία σε αυτό ήταν η κατηγοριοποίηση των στοιχείων μέτρησης. Πιο συγκεκριμένα, τα δεδομένα βρίσκονταν σε πολύ μεγάλο script και έπρεπε να βρεθεί ένας τρόπος να κατηγοριοποιηθούν.

5.2 Εργασίες

Οι εργασίες που θα μπορούσαν να υλοποιηθούν σχετίζονται τόσο με τα αποτελέσματα του μοντέλου όσο και με τον τρόπο παρουσίασής τους. Πιθανόν θα μπορούσε να υλοποιηθεί σε μεγαλύτερο χρονικό διάστημα για την βελτιστοποίηση του μοντέλου. Επιπλέον, θα μπορούσε να βρεθεί ένας τρόπος για την παρουσίαση των αποτελεσμάτων στους καλλιεργητές μέσω κάποιας εφαρμογής αντί της εμφάνισης του αποτελέσματος μέσω του υπολογιστή. Επιπλέον, θα μπορούσαν να χρησιμοποιηθούν και άλλες παράμετροι όπως το pH ή η συγκέντρωση χημικών στοιχείων των φυτών. Σε αυτή την περίπτωση οι διαστάσεις του μοντέλου θα ήταν περισσότερες. Τέλος, θα μπορούσε να δίνεται το μήνυμα για το τι χρειάζεται να γίνει για να 'ναι υγιής μέσω της χρήσης κάποιας οδηγίας.

Κώδικας

Machine Learning using K-Means

```
1 import numpy as np
2 import pandas as pd
3 import seaborn as sns
4 from sklearn.decomposition import PCA
5 from sklearn.preprocessing import StandardScaler
6 from sklearn import metrics
7 import numpy as np
8 from sklearn.model_selection import ParameterGrid
9 from sklearn.cluster import KMeans
10 import matplotlib.pyplot as plt
11 # We want to create a model that separates the days into
12 # clusters. The clusters would represent the different diseases. After
13 # the model is
14 # utilised we want to test if the program works effectively, so
15 # we will print the data from each cluster and check if they agree to
16 # similar diseases
17 # based on the lobe program.
18 # FOR INSTANCE
19 # MACHINE LEARNING MODEL          LOBE
20 # CLUSTER 5
21 # DAY 5                            DAY 5 --> YELLOW LEAF CURL
22 # DAY 13                          DAY 13 --> YELLOW LEAF CURL
23 # So, the cluster 5 represents the disease "Yellow Leaf Curl"
24 def loadData():
25     # Load dataset
26     home_data = pd.read_csv('C:/Users/marix/Downloads/data35.csv',
27                             usecols = ['Light', 'Temperature', 'Humidity' , 'Moisture' , ])
28     home_data.head()
```

```

25 print(home_data)
26
27 # Data shape
28 row,col = home_data.shape
29 print('there are {row} rows and {col} columns')
30 print(home_data.head(10))
31
32 # Work on the copy of data
33 home_data_scaled = home_data.copy()
34
35 #Scale Dataset
36 home_data_scaled[home_data_scaled.columns] =
StandardScaler().fit_transform(home_data_scaled)
37
38 print(home_data_scaled.describe() )
39
40
41
42
43     return home_data_scaled
44
45
46 def pca(df_scaled):
47
48 # Minimize the dimensions
49 pca_B = PCA(n_components = 2 )
50 pca_B_rslt = pca_B.fit_transform(df_scaled)
51 print('Explained variation per principal component:
{}'.format(pca_B.explained_variance_ratio_))
52 print('Cumulative variance explained by 2 principal components:
{:.2%}'.format(

```

```

53 np.sum(pca_B.explained_variance_ratio_)))
54
55
56 # Results from pca.components_
57 dataset_pca = pd.DataFrame(abs(pca_B.components_),
58                             columns=df_scaled.columns, index=['PC_1', 'PC_2'])
59
60 print('\n\n', dataset_pca)
61
62 print("Most important features")
63
64
65 print('As per PC 1:\n', (dataset_pca[dataset_pca >
66 0.3].iloc[0]).dropna())
67
68
69 print('\n\nAs per PC 2:\n', (dataset_pca[dataset_pca >
70 0.3].iloc[1]).dropna())
71
72 return pca_B_rslt, pca_B
73
74
75 def kmean_hyper_param_tuning(data):
76
77
78 # Given the data, the model selects the optimum number of
79 clusters according to the silhouette score
80
81 # The paramters can be as much as we want!
82 # parameters = [ 3, 9, 10, 15, 20, 25, 30, 35, 40]
83
84
85 parameters = [ 4 , 7, 10]
86
87 # Pass number of clusters as input
88

```

```

79
80
81     silhouette_scores += [ss]          # store all the scores
82
83     print('Parameter:', p, 'Score', ss)
84
85     # check p which has the best score
86     if ss > best_score:
87         best_score = ss
88         best_grid = p
89
90     # plotting silhouette score
91     plt.bar(range(len(silhouette_scores)),
92             list(silhouette_scores), align='center', color='#722f59', width=0.3)
93     plt.xticks(range(len(silhouette_scores)), list(parameters))
94     plt.title('Silhouette Score', fontweight='bold')
95     plt.xlabel('Number of Clusters')
96     plt.show()
97
98     return best_grid['n_clusters']
99
100
101
102
103 #def get_labels(data, centroids):
104 #     distances = centroids.apply(lambda x: np.sqrt(((data - x)
105 # ** 2).sum(axis=1))))
106 #     # return distances.idxmin(axis=1)

```

```

107
108 #def new_centroids(data, labels, k):
109 #     centroids = data.groupby(labels).apply(lambda x:
np.exp(np.log(x).mean())).T
110 #     return centroids
111
112
113
114
115 def visualizing_results(pca_result, label, centroids_pca):
116
117     # ----- Using Matplotlib for plotting-----
-----
118     x = pca_result[:, 0]
119     y = pca_result[:, 1]
120
121
122     plt.scatter(x, y, c=label, alpha=0.8, s=450) # plot
different colors per cluster
123     plt.title('Plant Health clusters')
124     plt.xlabel('PCA 1')
125     plt.ylabel('PCA 2')
126     print('the results are : \n ', pca_result ,label )
127     print(' \n ' )
128     print('the clusters are : ' , label)
129
130
131     print
132
133
134     print (" the centroids are : " , centroids_pca )
135     plt.scatter(centroids_pca[:, 0], centroids_pca[:, 1],
marker='x', s=55, linewidths=1.5,

```

```

136             color='red', edgecolors="black", lw=1.5)
137
138     plt.show()
139
140     max_iterations = 100
141     centroid_count = 4
142
143     new_label = label
144
145
146     old_centroids = pd.DataFrame
147
148     iteration = 1
149
150     print(old_centroids)
151     while iteration < max_iterations and centroids_pca !=
old_centroids:
152
153         old_centroids = centroids_pca
154         new_label = label
155         new_centroids = centroids_pca
156
157
158         visualizing_results(pca_result, label, centroids_pca)
159
160         iteration += 1
161
162
163     # labels = get_labels(pca_result, centroids_pca)

```

```

164         # centroids_pca = new_centroids(pca_result, labels,
centroid_count)
165         # visualizing_results(pca_result, labels, centroids_pca)
166
167
168
169
170 def main():
171
172     print("1. Loading Wine dataset\n")
173     data_scaled = loadData()
174
175     print("\n\n2. Reducing via PCA\n")
176     pca_result, pca_B = pca(data_scaled)
177
178     print("\n\n3. HyperTuning the Parameter for KMeans\n")
179     optimum_num_clusters = kmean_hyper_param_tuning(data_scaled)
180     print("optimum num of clusters =", optimum_num_clusters)
181
182     # fitting KMeans
183     kmeans = KMeans(n_clusters=optimum_num_clusters)
184     kmeans.fit(data_scaled)
185     centroids = kmeans.cluster_centers_
186     centroids_pca = pca_B.transform(centroids)
187
188     print("\n\n4. Visualizing the data")
189     visualizing_results(pca_result, kmeans.labels_,
centroids_pca)
190
191
192
193

```



```
194
195
196
197 if __name__ == "__main__":
198     main()
```

WEB

```
1 import re
2 import bs4
3 from bs4 import BeautifulSoup
4 import requests
5 import pandas as pd
6 from datetime import datetime
7 from datetime import date, datetime
8 import itertools
9 import numpy as np
10
11 url="https://www.meteo.gr/cf.cfm?city_id=1"
12 page = requests.get(url)
13 print(page)
14 soup = BeautifulSoup(page.content, 'html.parser')
15 print(soup)
16
17
18 table = soup.find_all("table")
19 y_data = [row.text.splitlines() for row in table]
20 y_data = y_data[:-8]
21 for i in range(len(y_data)):
22     y_data[i] = y_data[i][2:len(y_data[i]):1]
23 print("y=",y_data)
24
25
26 Dates_r = pd.date_range(start = "25/08/2023",end =
"27/08/2023",freq = "d")
27 dates = [str(i)[:4] + str(i)[5:7] for i in Dates_r]
28 dates[0:5]
29 for k in range(len(dates)):
```

```

30
31 url = "https://www.meteo.gr/cf.cfm?city_id=1"
32 # print(dates[k])
33 #for u in url:
34 for i in range(2):
35                                     v = {'text':
[ '.'.join(re.findall("\d+",str(y_data[i][j].split()[:3])))for j in
range(len(y_data[i]))] }
36
37
38         df = pd.DataFrame(v )
39
40
41         df = df.apply(pd.to_numeric, errors = 'coerce')
42         df = df.dropna()
43         print(df)
44         num = df.values.tolist()
45         humidity1 = df.values.tolist()
46         humidity2 = df.values.tolist()
47         wind = df.values.tolist()
48         temp = df.values.tolist(
49         print(num[i:len(num):38])
50         print("\n")
51
52         print("the humidity is: ")
53
54 b = [0 for i in range(len(humidity1))]
55 q = [9 for i in range(800)]
56 num = 0
57
58 for i in range (len(humidity1) ):

```

```

58
59         if humidity1[i-1] == humidity1[i]:
60             b[i] = humidity1[i]
61
62             value = float(''.join(map(str,humidity1[i])))
63             b[i] =value
64
65             #q[i] = [np.add(b[i],b[i-1])]
66             if value > 9:
67                 b[i] = value
68                 print(b[i])
69                 num = num + 1
70 d97 = sum(b)/num
71 print("the aner is : ",d97)
72
73 temp = [0 for i in range(len(humidity1))]
74 num8 = 0
75
76 for i in range (len(humidity1)):
77     if humidity1[i-1] == humidity1[i]:
78         temp[i] = humidity1[i-2]
79
80         value32 = float(''.join(map(str,humidity1[i-
81 2]])))
82         temp[i] = value32
83
84         if value32 > 9:
85             temp[i] = value32
86             num8 = num8 + 1
87
88             print(temp[i])

```

```

88
89 d98 = sum(temp)/ num8
90 print("the aver is", d98)
91
92 wind1 = [0 for i in range(len(humidity1))]
93 num93 = 0
94 for i in range (len(humidity1)):
95         if humidity1[i-1] == humidity1[i]:
96                 wind1[i] = humidity1[i + 2]
97
98                 value79 = float(''.join(map(str,humidity1[i +
99                 wind1[i] = value79
100
101                 if num93<30:
102                         wind1[i] = value79
103                         print(wind1[i])
104                         num93 = num93 +1
105 d35 = sum(wind1)/num93
106 print("the result is :", d35)
107 print("the is")
108
109 # HEALTHY
110 if d97 > 65.0 and d97 < 75.0 and d98 > 33.0 and d98 <34.0:
111         print("The plant is going to be healthy for the next 3
112         days")
113
114 # LEAF MOLD
115 if d97 > 52.0 and d97 < 60.0 and d98 > 26.0 and d98 < 27.0:
116         print("The plant might have leaf mold in the next 3 days")

```

```

117
118
119
120 -# LATE BLIGHT
121 if d97 > 58.0 and d97 < 63.0 and d98 > 19.0 and d98 < 26.0:
122     print(" The plant might have late blight ")
123
124 # TARGET SPOT
125 if d97 > 58.0 and d97 < 75.0 and d98 > 23.0 and d98 < 32.0:
126     print(" The plant might have target spot or Septoria leaf
spot or early blight or bacterial spot in the next 3 days")
127 # BACTERIAL SPOT
128 if d97 > 48.0 and d97 < 75.0 and d98 > 24.0 and d97 < 30.0:
129     print("The plant might have bacterial spot target spot
early blight or Septoria leaf spot in the next 3 days")
130
131
132 # SEPTORIA LEAF SPOT
133 if d97 == 75.0 and d98 == 23.0:
134     print("The plant might have septoria leaf spot in the next
3 days")
135
136
137
138 # EARLY BLIGHT
139 if d97 > 48.0 and d97 < 75.0 and d98 > 24.0 and d98 < 75.0:
140     print("The plant might have early blight or target spot or
bacterial spot")
141
142 # SPIDER MITES
143 if d97 == 35.0 and d98 == 34.0:
144     print("The plant might have spider mites in the next 3
days")

```

```
145 # YELLOW LEAF CURL
146 if d97 > 52.0 and d97 < 75.0 and d98 > 21.0 and d98 < 32.0:
147     print("The plant might have yellow leaf curl in the next 3
days" )
```

Βιβλιογραφία

- [1] Nipuna Chamara, Md Didarul Islam, Geng (Frank) Bai, Yeyin Shi, Yufeng Ge, Ag-IoT for crop and environment monitoring: Past, present, and future, *Agricultural Systems*, Volume 203, 2022, 103497, ISSN 0308-521X, <https://doi.org/10.1016/j.agsy.2022.103497>
- [2] Melanie McCaig, Davar Rezaia, Rozita Dara, Is the Internet of Things a helpful employee? An exploratory study of discourses of Canadian farmers, *Internet of Things*, Volume 17, 2022, 100466, ISSN 2542-6605, <https://doi.org/10.1016/j.iot.2021.100466>.
- [3] Hooi Ren Lim, Kuan Shiong Khoo, Wen Yi Chia, Kit Wayne Chew, Shih-Hsin Ho, Pau Loke Show, Smart microalgae farming with internet-of-things for sustainable agriculture, *Biotechnology Advances*, Volume 57, 2022, 107931, ISSN 0734-9750, <https://doi.org/10.1016/j.biotechadv.2022.107931>
- [4] Online, available at <https://robu.in/5-pros-and-5-cons-of-raspberry-pi/> accessed on 28/08/23
- [5] Online, available at <https://socialcompare.com/en/comparison/raspberrypi-models-comparison> accessed on 28/08/23
- [6] Online, available at <https://github.com/Seeed-Studio/grove.py/tree/master/grove> accessed on 07/09/23
- [7] Online, available at <https://www.makeuseof.com/raspberry-pi-3b-plus-still-worth-it-2023/> accessed on 28/09/23
- [8] Online, available at https://wiki.seeedstudio.com/Grove_Base_Kit_for_Raspberry_Pi/ accessed on 07/09/23
- [9] Online, available at <https://vector4engineering.com/product/nodemcu-esp8266/> accessed on 18/09/23
- [10] Online, available at <https://www.pixelspot.net/2017/09/26/depth-understanding-cortex-a53-mobile-socs/> accessed on 08/09/23
- [11] Online, available at https://apps.lucidcentral.org/ppp/text/web_full/entities/tomato_target_spot_163.html/ accessed on 18/09/23
- [12] Online, available at <https://ipm.cahn.uconn.edu/bacterial-leaf-spot-in-peppers/> accessed on 16/09/23
- [13] Online, available at <http://extension.msstate.edu/publications/bacterial-speck-and-bacterial-spot-tomatoes> accessed on 18/09/23
- [14] Online, available at <https://www.missouribotanicalgarden.org/gardens-gardening/your-garden/help-for-the-home-gardener/advice-tips-resources/pests-and-problems/diseases/fungal-spots/septoria-leaf-spot-of-tomato> accessed on 18/09/23
- [15] Online, available at <https://www.youtube.com/watch?v=GRkC8oHUMYA&t=4s> accessed on 18/09/23

[16] Online, available at <https://content.ces.ncsu.edu/tomato-late-blight> accessed on 18/09/23

[17] Budai, Bettina & Tóth, Ambrus & Borsos, Petra & Frank, Veronica & Shariati, Sonaz & Fejér, Bence & Folhoffer, Aniko & Szalay, Ferenc & Bérczi, Viktor & Kaposi, Pál. (2020). Three-dimensional CT texture analysis of anatomic liver segments can differentiate between low-grade and high-grade fibrosis. BMC medical imaging. 20. 108. 10.1186/s12880-020-00508-w.

[18] Online, available at <https://www.gardeningchannel.com/tomato-diseases-how-to-fight-leaf-mold/> accessed on 18/09/23

[19] Online, available at <https://www.lobe.ai/> accessed on 18/09/23

[20] Online, available at https://extension.usu.edu/pests/ipm/notes_ag/veg-TMV-ToMV , accessed on 18/09/23

[21] Online, available at https://www.youtube.com/watch?v=6db7a5Hum_I accessed on 18/09/23

[22] Online, available at https://scikit-learn.org/stable/modules/generated/sklearn.metrics.silhouette_score.html accessed on 16/10/23