



ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΛΟΠΟΝΝΗΣΟΥ
ΣΧΟΛΗ ΜΗΧΑΝΙΚΩΝ
ΤΜΗΜΑ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ
ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

**Δημιουργία chatbot με χρήση αλγορίθμων μηχανικής μάθησης και
βελτιστοποίηση για χρήση σε ενσωματωμένη υπολογιστική
πλατφόρμα**

Σταυρόπουλος Βασίλειος (2924)

Επιβλέπων: Νικόλαος Βώρος

Συν επιβλέπουσα: Ευανθία Φαλιάγκα

ΠΑΤΡΑ 2022

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή

Πάτρα, Ημερομηνία

ΕΠΙΤΡΟΠΗ ΑΞΙΟΛΟΓΗΣΗΣ

1. Ονοματεπώνυμο, Υπογραφή
2. Ονοματεπώνυμο, Υπογραφή
3. Ονοματεπώνυμο, Υπογραφή

Υπεύθυνη Δήλωση Φοιτητή

Βεβαιώνω ότι είμαι συγγραφέας αυτής της εργασίας και ότι κάθε βοήθεια την οποία είχα για την προετοιμασία της είναι πλήρως αναγνωρισμένη και αναφέρεται στην εργασία. Επίσης έχω αναφέρει τις όποιες πηγές από τις οποίες έκανα χρήση δεδομένων, ιδεών ή λέξεων, είτε αυτές αναφέρονται ακριβώς είτε παραφρασμένες. Επίσης βεβαιώνω ότι αυτή η εργασία προετοιμάστηκε από εμένα προσωπικά ειδικά για τη συγκεκριμένη εργασία.

Η έγκριση της διπλωματικής εργασίας από το Τμήμα Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών του

Πανεπιστημίου Πελοποννήσου δεν υποδηλώνει απαραίτητως και αποδοχή των απόψεων του συγγραφέα εκ μέρους του Τμήματος.

Η παρούσα εργασία αποτελεί πνευματική ιδιοκτησία του φοιτητή Σταυρόπουλου Βασίλειου που την εκπόνησε.

Στο πλαίσιο της πολιτικής ανοικτής πρόσβασης ο συγγραφέας/δημιουργός εκχωρεί στο Πανεπιστήμιο Πελοποννήσου, μη αποκλειστική άδεια χρήσης του δικαιώματος αναπαραγωγής, προσαρμογής, δημόσιου δανεισμού, παρουσίασης στο κοινό και ψηφιακής διάχυσής τους διεθνώς, σε ηλεκτρονική μορφή και σε οποιοδήποτε μέσο, για διδακτικούς και ερευνητικούς σκοπούς, άνευ ανταλλάγματος και για όλο το χρόνο διάρκειας των δικαιωμάτων πνευματικής ιδιοκτησίας. Η ανοικτή πρόσβαση στο πλήρες κείμενο για μελέτη και ανάγνωση δεν σημαίνει καθ' οιονδήποτε τρόπο παραχώρηση δικαιωμάτων διανοητικής ιδιοκτησίας του συγγραφέα/δημιουργού ούτε επιτρέπει την αναπαραγωγή, αναδημοσίευση, αντιγραφή, αποθήκευση, πώληση, εμπορική χρήση, μετάδοση, διανομή, έκδοση, εκτέλεση, «μεταφόρτωση» (downloading), «ανάρτηση» (uploading), μετάφραση, τροποποίηση με οποιονδήποτε τρόπο, τμηματικά ή περιληπτικά της εργασίας, χωρίς τη ρητή προηγούμενη έγγραφη συναίνεση του συγγραφέα/δημιουργού. Ο συγγραφέας/δημιουργός διατηρεί το σύνολο των ηθικών και περιουσιακών του δικαιωμάτων.

Περίληψη

Σκοπός της παρούσας διπλωματικής εργασίας ήταν, να δημιουργηθεί ένας προσωπικός βοηθός στην ελληνική γλώσσα που θα μπορεί να αλληλεπιδρά με το χρήστη με όσο το δυνατόν πιο φυσικό τρόπο.

Ο προσωπικός βοηθός που αναπτύχθηκε ενσωματώθηκε σε ένα περιβάλλον έξυπνης οικίας (AAL HOUSE) και μπορεί να επικοινωνήσει με το χρήστη για γενικά θέματα. Μερικά από αυτά είναι τι καιρό θα έχει σήμερα ή όλη την εβδομάδα, ποια είναι τα τελευταία αθλητικά νέα καθώς μπορεί και να απαντά σε ερωτήσεις λαμβάνοντας υπόψη τις τιμές των **αισθητήρων του σπιτιού**.

Πιο αναλυτικά, στο πρώτο κεφάλαιο γίνεται αναφορά το τι είναι ένα chatbot και ποια είναι η ιστορία τους ανά τα χρόνια.

Στην συνέχεια, αναλύονται τα υποβοηθούμενα περιβάλλοντα και οι τεχνολογίες που υπάρχουν μέσα σε ένα έξυπνο σπίτι.

Στο κεφάλαιο 3 και 4 παρουσιάζεται πως λειτουργούσε η Αλεξάνδρα πριν και πως τώρα με την ανάλυση του κώδικας που έχει προστεθεί.

Τέλος, αναφέρονται μελλοντικές επεκτάσεις που μπορούν να υπάρξουν καθώς επίσης και τα συμπεράσματα που προέκυψαν στην παρούσα διπλωματική εργασία.

Abstract

The aim of this thesis was to create a personal assistant in the Greek language that can interact with the user in the most natural way possible.

The developed personal assistant was integrated into a smart home environment (AAL HOUSE) and can communicate with the user on general issues. Some of these are what the weather will be like today or throughout the week, what the latest sports news is as well as it can also answer questions taking into account the values of the home's sensors.

In more detail, the first chapter discusses what a chatbot is and what their history is over the years.

Then, the assisted environments and the technologies present within a smart home are discussed.

Chapter 3 and 4 shows how Alexandra worked before and how it works now with the analysis of the code that has been added.

Finally, future extensions that can be made are mentioned as well as the conclusions drawn in this thesis.

Πίνακας περιεχομένων

Περίληψη	3
Abstract	4
Κεφάλαιο 1: Εισαγωγή	9
1.1: Τι είναι τα chatbots	9
1.2: Η ιστορία των chatbot	9
1.3 Περιοχές εφαρμογής των chatbots	16
1.4 Πλεονεκτήματα των chatbots	16
1.5 Μειονεκτήματα των chatbots	16
Κεφάλαιο 2: Χρήση των chatbot σε υποβοηθούμενα περιβάλλοντα διαβίωσης	16
2.1 Τι είναι τα υποβοηθούμενα περιβάλλοντα διαβίωσης	16
2.2 Παρουσίαση της έξυπνης οικίας	17
Κεφάλαιο 3: Πως λειτουργεί η Kalliope (Αλεξάνδρα)	21
3.1 Πως λειτουργούσε πριν	23
3.2 Πως λειτουργεί τώρα	26
Κεφάλαιο 4: Υλοποίηση	Σφάλμα! Δεν έχει οριστεί σελιδοδείκτης.
4.1 Τι είναι η γλώσσα προγραμματισμού Python	46
4.2 Τι είναι η βιβλιοθήκη Scikit-learn	46
4.3 Πως λειτουργεί ο κώδικας	47
4.4 Τι είναι το μοντέλο bag of words	48
4.4 Ανάλυση κώδικα του κυρίως προγράμματος	49
4.5 Πως μπορούμε να προσθέσουμε καινούργιες προτάσεις	53
4.6 Ανάλυση κώδικα του προγράμματος ένταξης προτάσεων	54
Κεφάλαιο 5: Συμπεράσματα και μελλοντικές επεκτάσεις	57
Αναφορές	58

Πίνακας Εικόνων

- [Εικόνα 1](#)
- [Εικόνα 2](#)
- [Εικόνα 3](#)
- [Εικόνα 4](#)
- [Εικόνα 5](#)
- [Εικόνα 6](#)
- [Εικόνα 7](#)
- [Εικόνα 8](#)
- [Εικόνα 9](#)
- [Εικόνα 10](#)
- [Εικόνα 11](#)
- [Εικόνα 12](#)
- [Εικόνα 13](#)
- [Εικόνα 14](#)
- [Εικόνα 15](#)
- [Εικόνα 16](#)
- [Εικόνα 17](#)
- [Εικόνα 18](#)
- [Εικόνα 19](#)
- [Εικόνα 20](#)
- [Εικόνα 21](#)
- [Εικόνα 22](#)
- [Εικόνα 23](#)
- [Εικόνα 24](#)
- [Εικόνα 25](#)
- [Εικόνα 26](#)
- [Εικόνα 27](#)
- [Εικόνα 28](#)
- [Εικόνα 29](#)
- [Εικόνα 30](#)
- [Εικόνα 31](#)
- [Εικόνα 32](#)
- [Εικόνα 33](#)
- [Εικόνα 34](#)
- [Εικόνα 35](#)
- [Εικόνα 36](#)

- [Εικόνα 37](#)
- [Εικόνα 38](#)
- [Εικόνα 39](#)
- [Εικόνα 40](#)
- [Εικόνα 41](#)
- [Εικόνα 42](#)
- [Εικόνα 43](#)
- [Εικόνα 44](#)
- [Εικόνα 45](#)
- [Εικόνα 46](#)
- [Εικόνα 47](#)
- [Εικόνα 48](#)
- [Εικόνα 49](#)
- [Εικόνα 50](#)
- [Εικόνα 51](#)
- [Εικόνα 52](#)
- [Εικόνα 53](#)
- [Εικόνα 54](#)

Πίνακας 1

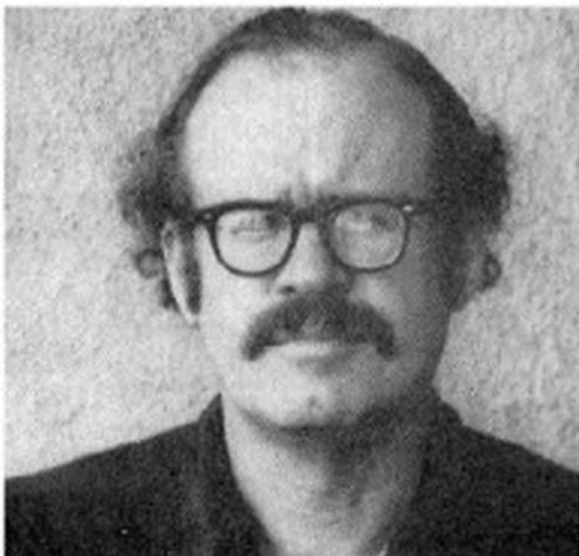
Πίνακας Σχημάτων

- [Πίνακας 1](#)
- [Πίνακας 2](#)
- [Πίνακας 3](#)
- [Πίνακας 4](#)
- [Πίνακας 5](#)
- [Πίνακας 6](#)
- [Πίνακας 7](#)
- [Πίνακας 8](#)
- [Πίνακας 9](#)
- [Πίνακας 10](#)
- [Πίνακας 11](#)
- [Πίνακας 12](#)
- [Πίνακας 13](#)
- [Πίνακας 14](#)
- [Πίνακας 15](#)
- [Πίνακας 16](#)
- [Πίνακας 17](#)
- [Πίνακας 18](#)
- [Πίνακας 19](#)
- [Πίνακας 20](#)

Η δυνατότητα επικοινωνίας της ήταν περιορισμένη αλλά ήταν πηγή έμπνευσής για να δημιουργηθούν μελλοντικά chatbots. Η ELIZA ανάλυε της λέξεις της οποίες λάμβανε και απαντούσε με βάζει κάποιους συγκεκριμένους κανόνες.

Το επόμενο chatbot ήταν το PARRY [6] το οποίο δημιουργήθηκε από το ψυχίατρο Kenneth Colby στο Πανεπιστήμιο του Stanford το 1972 .

Το συγκεκριμένο chatbot είχε ως λειτουργία το να προσομοίωση έναν άνθρωπο με σχιζοφρένεια.

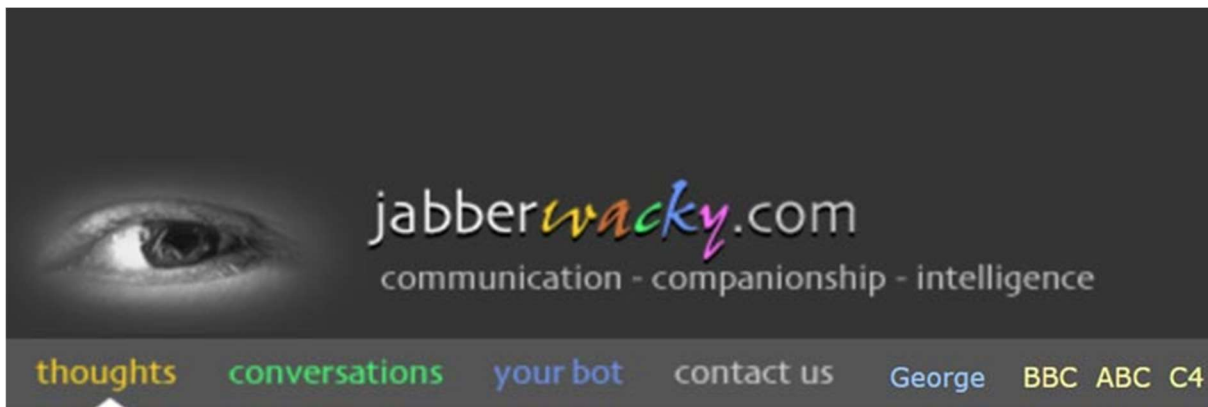


```
H: Good afternoon. My name is  
  Doctor Jones.  
P: Pleased to meet you.  
H: How are you?  
P: All right I guess.  
H: Could you tell me your name?  
P: Pat Smith.  
H: I though you might have some  
  problems.  
P: I've been upset lately.  
H: By what?  
P: People get on my nerves  
  sometimes
```

Εικόνα 2

Το πρώτο chatbot με Τεχνητή νοημοσύνη δημιουργήθηκε το 1988 από των Rollo Carpenter έναν βρετανό προγραμματιστή και ονομάστηκε Jabberwacky [7].

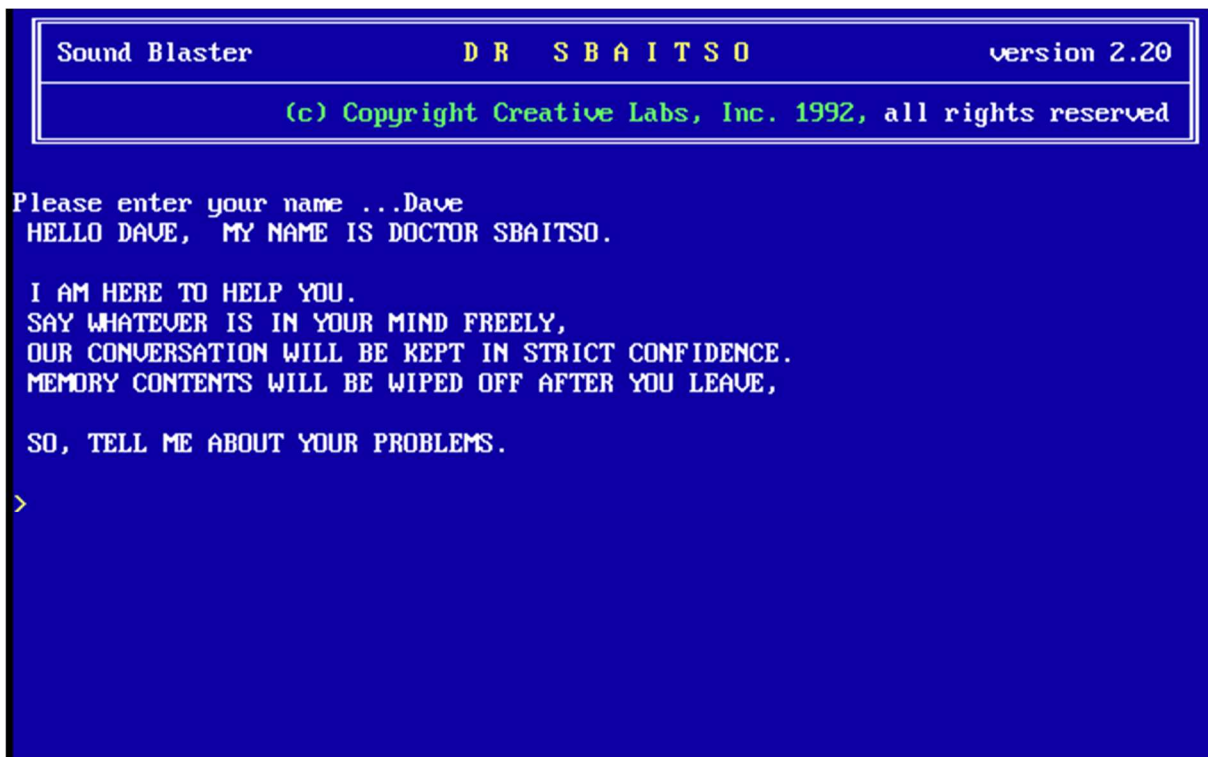
Αυτό το chatbot απαντούσε με βάση τα συμφραζόμενα για να μπορεί να απαντάει σε προηγούμενες συνομιλίες.



Εικόνα 3

To 1992 δημιουργήθηκε το chatbot με όνομα Dr. Sbaitso [8](Sound Blaster Artificial Intelligent Text to Speech Operator) από την Creative Labs στην Σιγκαπούρη.

Σαν στόχο είχε να εξομοιώσει την λειτουργία της φωνής για να μπορεί να επικοινωνήσει με το χρήστη σαν να ήταν ψυχολόγος .



Εικόνα 4

Το επόμενο βήμα για τα chatbot ήταν η ALICE [9](Artificial Linguistic Internet Computer Entity) δημιουργήθηκε το 1995 από τον Richard Wallace . Λειτουργούσε με παρόμοιο τρόπο με την ELIZA αλλά ήταν στο διαδίκτυο οπότε μπορούσε να προσφέρει περισσότερα θέματα για να συζητήσει ο χρήστης.



Εικόνα 5

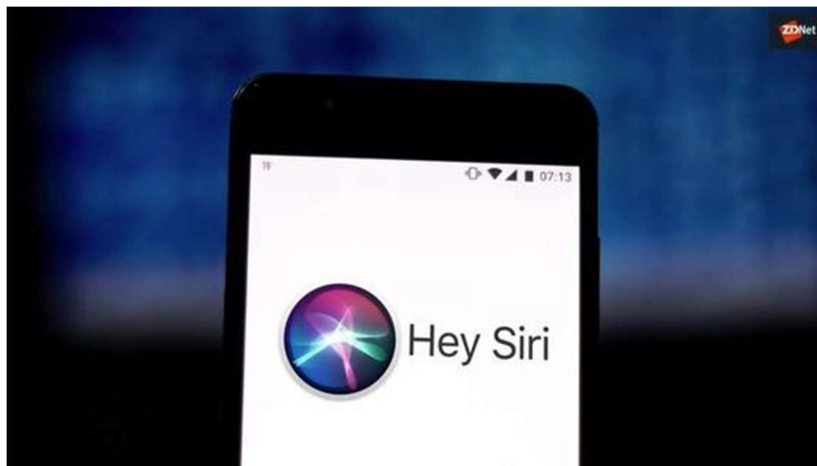
Το επόμενο chatbot που δημιουργήθηκε από την ActiveBuddy ήταν το SmarterChild [10] το 2001 το οποίο ήταν διαθέσιμο σε διάφορα messenger της εποχής όπως το America Online (AOL) και το Microsoft (MSN).

Θεωρείται ένα από τα πρώτα chatbot τα οποία βοηθούσαν τον χρήστη σε διάφορα καθημερινά καθήκοντα όπως να φέρει πληροφορίες για τον καιρό , για τα νέα και πολλά περισσότερα.



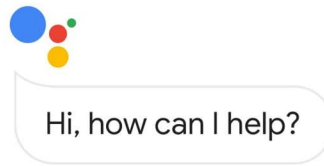
Εικόνα 6

Το 2010 η APPLE δημιουργείσai την Siri [1] στην οποία οι χρήστες μπορούσαν να πραγματοποιήσουν ερωτήσεις και συνομιλίες μέσω φωνητικών εντολών . Η Siri μπορεί με τη χρήση μιας σύνδεσης στο διαδίκτυο να κάνει συστάσεις στον χρήστη σε διάφορα θέματα.



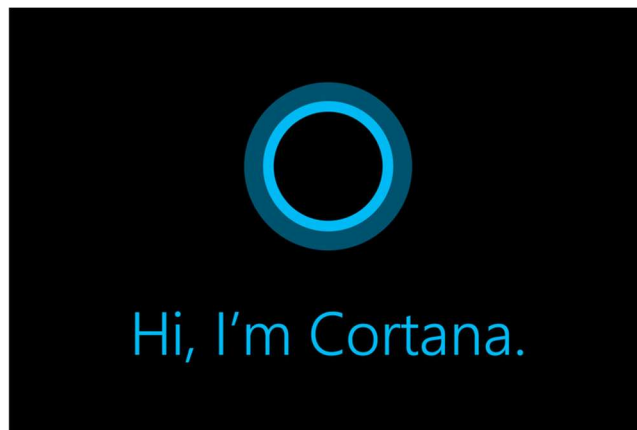
Εικόνα 7

Η Google το 2012 δημιούργησε το δικό της Chatbot το Google Assistant[2] το οποίο μπορεί να λάβει υπόψη την ημέρα, την περιοχή και της προτίμησης που είχε δηλώσει ο χρήστης.



Εικόνα 8

Το 2014 η Microsoft έφτιαξε το δικό της Chatbot το οποίο ονομάστηκε Cortana [3]. Μπορούσε να αναγνωρίσει τις φωνητικές εντολές και να πραγματοποιήσει διάφορες εντολές όπως να γράψει και να διαβάσει την ηλεκτρονική αλληλογραφία, να γράψει λίστες και να διαβάσει.



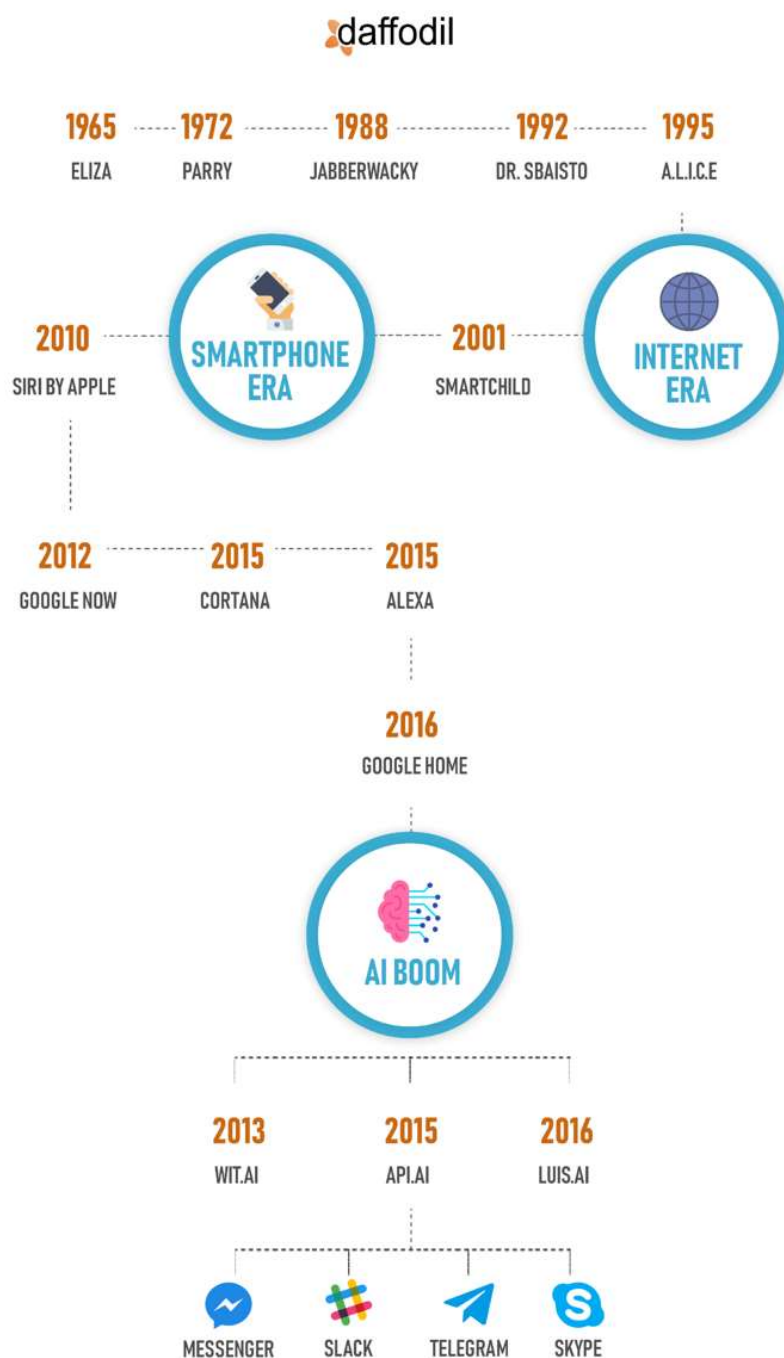
Εικόνα 9

Τον ίδιο χρόνο η Amazon κατασκεύασε την Alexa [11] η οποία μπορούσε να χειριστεί συσκευές στο διαδίκτυο των πραγμάτων. [12]



Εικόνα 10

Στην παρακάτω εικόνα παρουσιάζεται η ιστορική εξέλιξη των chatbot που αναλύθηκε προηγουμένως.



Εικόνα 11

1.3 Περιοχές εφαρμογής των chatbots

Τα chatbot μπορούν να χρησιμοποιηθούν σε πολλούς τομείς, κάποιοι από αυτούς είναι :

- Ο επαγγελματικός τομέας
- Ο τομέας υγείας
- και ο ατομικός τομέας

1.4 Πλεονεκτήματα των chatbots

Τα πλεονεκτήματα ενός chatbot είναι τα εξής:

- Μείωση ανθρώπινου δυναμικού
- Γρήγορη εξυπηρέτηση
- Διαθέσιμο οποιαδήποτε στιγμή

1.5 Μειονεκτήματα των chatbots

Τα μειονεκτήματα ενός chatbot είναι τα εξής:

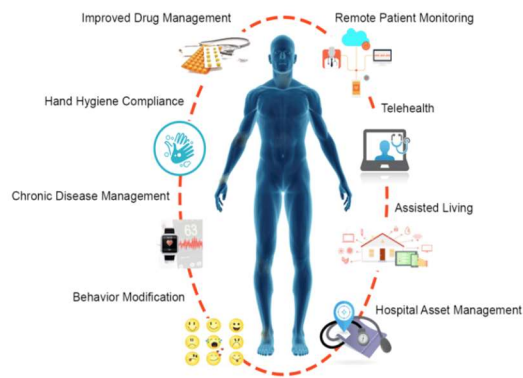
- Περιορισμένο σύνολο απαντήσεων
- Δεν είναι πάντα σωστά στις απαντήσεις τους

Κεφάλαιο 2: Χρήση των chatbot σε υποβοηθούμενα περιβάλλοντα διαβίωσης

Σε αυτό το κεφάλαιο θα αναλύσουμε τι είναι ένα υποβοηθούμενο περιβάλλον διαβίωσης και ποια είναι χρήση ενός chatbot σε αυτό.

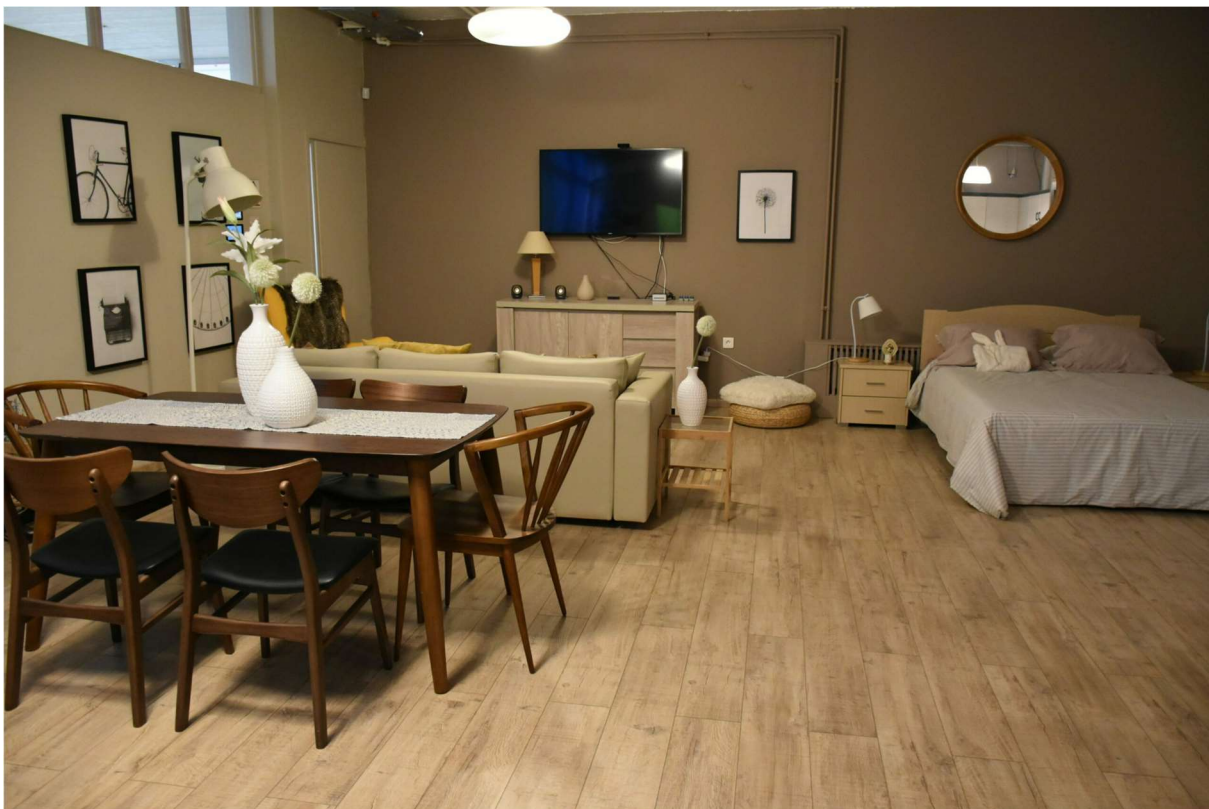
2.1 Τι είναι τα υποβοηθούμενα περιβάλλοντα διαβίωσης

Ένα υποβοηθούμενο περιβάλλον διαβίωσης (AAL) περιλαμβάνει τη χρήση συσκευών και μεθόδων για να διασφαλιστεί ότι οι ηλικιωμένοι στο σπίτι παραμένουν ασφαλείς και ανεξάρτητη. Ο χώρος ενός υποβοηθούμενου περιβάλλοντος διαβίωσης (AAL) περιλαμβάνει έξυπνες συσκευές, ασύρματα δίκτυα, εφαρμογές λογισμικού, υπολογιστές και ιατρικούς αισθητήρες. Αυτές οι τεχνολογίες επιτρέπουν σε έναν ηλικιωμένο να μπορέσει να ζήσει πιο ανεξάρτητα και πιο ασφαλής με το να μπορέσουν να ειδοποιήσουν έγκυρα τον Γιατρό τους ή κάποια άτομα που έχουν ορίσει ως πρόσωπα έκτακτης ανάγκης. [14]



Εικόνα 12

2.2 Παρουσίαση της έξυπνης οικίας



Εικόνα 13

Η έξυπνη οικία έχει υλοποιηθεί από τα μέλη του Εργαστηρίου Σχεδιασμού Ενσωματωμένων Συστημάτων και Εφαρμογών, του τμήματος Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών του Πανεπιστημίου Πελοποννήσου.

Το οικοσύστημα της έξυπνης οικίας περιλαμβάνει διάφορες υπηρεσίες, αυτοματισμούς και τεχνολογίες για την διευκόλυνση των ηλικιωμένο ανθρώπων.

Πιο συγκεκριμένα, κάποιες από τις τεχνολογίες που περιέχει η έξυπνη οικία είναι :



Εικόνα 14

- Διάφοροι αισθητήρες που μέσω της τεχνολογίας MQTT μπορούν να μας επιστρέφουν σε πραγματικό χρόνο τη θερμοκρασία την υγρασία το βαρομετρικό καθώς επίσης και την φωτεινότητα κάθε δωματίου της έξυπνης οικίας και να τις εμφανίζουν μαζί με την σε διάφορα πάνελ που είναι τοποθετημένα κατά το μήκος της οικίας.

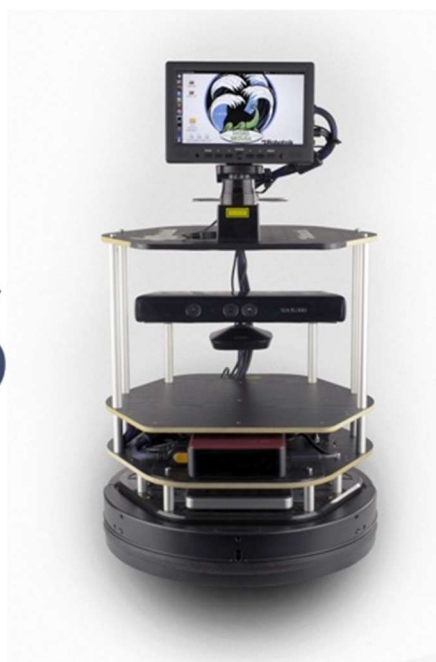
Μια ακόμη λειτουργία που απεικονίζεται στα πάνελ είναι η κατανάλωση ενέργειας.

- Συσκευή ελέγχου που χρησιμοποιεί το πρωτόκολλο KNX. Μέσω αυτής της συσκευής μπορούμε να ελέγξουμε τα φώτα του σπιτιού και της ηλεκτρονικές συσκευές τις οποίες περιέχει η οικία. Σε αυτή τη συσκευή έχουν ενταχθεί διάφορα σενάρια με τα οποία μπορούμε να εκτελέσουμε ώστε να ελέγξουμε πολλές συσκευές ταυτόχρονα.



Εικόνα 15

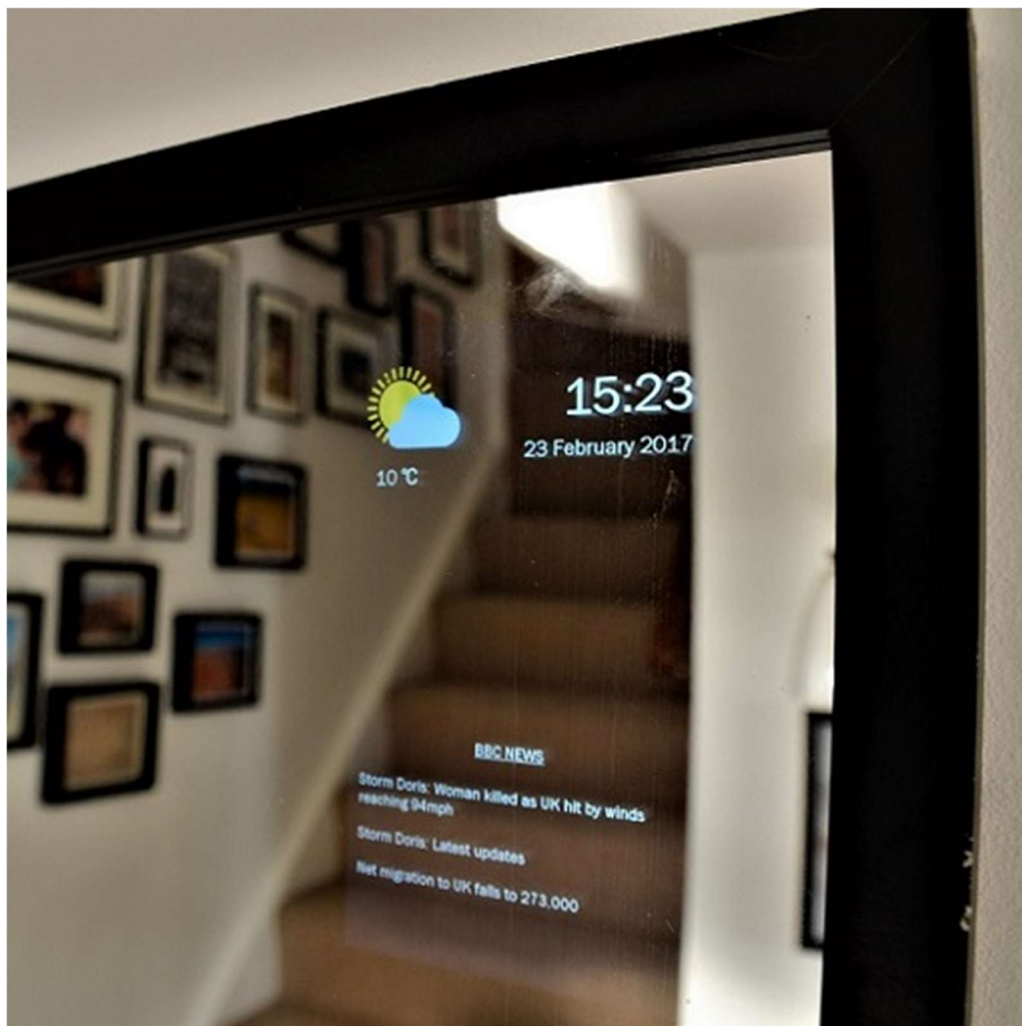
ROS



Εικόνα 16

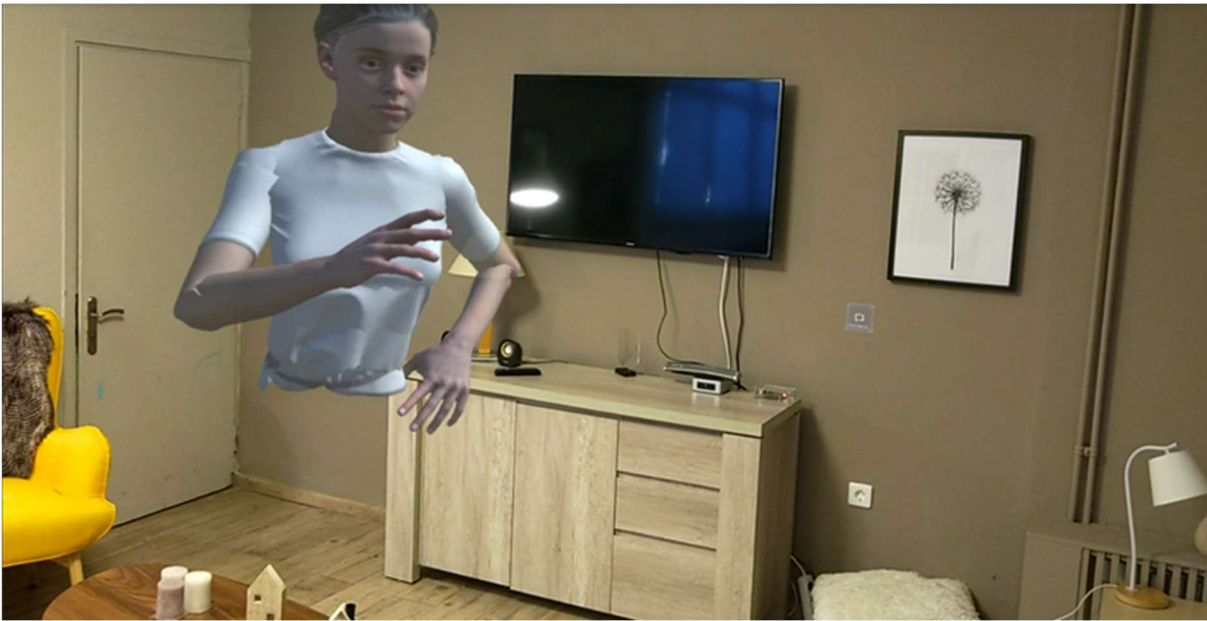
- Η οικία περιέχει έναν ρομποτικός βοηθός τύπου Turtlebot2. Πιο αναλυτικά, ο ρομποτικός βοηθός που αναφέρθηκε προηγουμένως χρησιμοποιεί ένα laser scanner για την αναγνώριση του χώρου καθώς επίσης μπορεί να αναγνωρίσει διάφορα εμπόδια

υπάρχουν στον χώρο τον οποίο έχει γίνει χαρτογράφηση. Επίσης, χρησιμοποιεί μια στερεοσκοπική κάμερα.



Εικόνα 17

- Στην έξυπνη οικία υπάρχει ο έξυπνος καθρέφτης που χρησιμοποιεί την εφαρμογή magic mirror. Είναι μία συσκευή που μπορεί και συλλέγει διάφορες βιομετρικές πληροφορίες όπως είναι οι καρδιακοί παλμοί και η θερμοκρασία του ατόμου που κάθεται μπροστά από τον έξυπνο καθρέφτη. Επίσης, μπορεί να μας εμφανίσει πληροφορίες όπως επίκαιρα νέα, καιρικές προβλέψεις και να θυμίσει στον ηλικιωμένο να πάρει το φάρμακο του το οποίο του έχει συνταγογραφήσει ο προσωπικός του γιατρός.



Εικόνα 18

- Εν συνεχεία, περιέχει Σύστημα Φωνητικής Αλληλεπίδρασης στα Ελληνικά με Χρήση Τεχνητής Νοημοσύνης με την ονομασία Αλεξάνδρα που χρησιμοποιεί το open source framework kalliope. Μέσω της Αλεξάνδρας μπορούμε να ελέγξουμε τα φώτα του σπιτιού μέσω http request, ελέγξουμε τη θερμοκρασία του σπιτιού, να χειριστούμε το ρομποτικό βοηθό και να πάρουμε διάφορες πληροφορίες μέσω του διαδικτύου όπως τον καιρό και τα νέα της επικαιρότητας.

Κεφάλαιο 3: Πως λειτουργεί η Kalliope (Αλεξάνδρα)

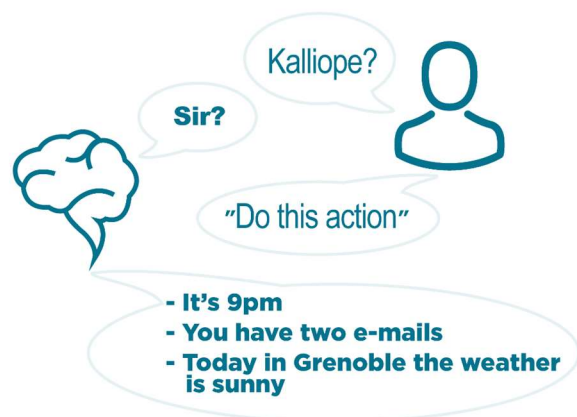
CONFIGURE YOUR PERSONAL ORDERS

Use neurons as plugins and build Kalliope's brain with an action of your choice

"Do this action"

USE YOUR ASSISTANT - KALLIOPE

Following your previous configuration, enjoy the voice-controlled system to get a clear and custom answer to your order



Η Kalliope είναι ένα framework που βοήθα στη δημιουργία ενός ψηφιακού προσωπικού βοηθού.

Η ιδέα είναι να ότι δημιουργούμε έναν εγκέφαλο συνδέοντας σήματα εισόδου (φωνητικές εντολές, προγραμματισμένα συμβάντα, μηνύματα MQTT, συμβάντα GPIO κ.λπ.) με μία ή περισσότερες ενέργειες που ονομάζονται νευρώνες. Δημιουργούμε ένα chatbot επιλέγοντας και συναρμολογώντας υπάρχοντες νευρώνες χωρίς να χρειαστεί να γράψουμε καθόλου κώδικα. Ωστόσο, αν χρειαστούμε μια συγκεκριμένη ενότητα, μπορούμε να τη γράψουμε μόνοι μας, να την προσθέσετε στο έργο μας και να τη συστήνουμε στην κοινότητα.[13]

Ένα απλό παράδειγμα στο πως λειτουργεί είναι το εξής :

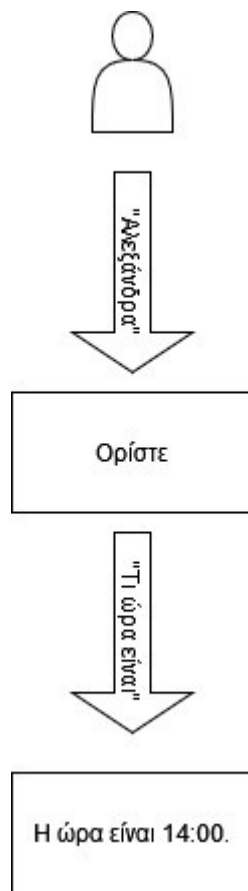
```
- name: "Hello-world"
  signals:
    - order: "say hello"
  neurons:
    - say:
      message: "Hello world!"
```

Πίνακας 2

Έχουμε το όνομά του neuron το οποίο το ονομάζουμε "Hello-world" στη συνέχεια δηλώνουμε τη φωνητική εντολή "say hello" και τέλος δηλώνουμε το τι θα κάνει αυτός ο νευρώνας σε αυτή την περίπτωση απλά εμφανίζει και λέει ένα μήνυμά "Hello world!".

Γενικότερα η ροή για να χρησιμοποιήσουμε την kalliope είναι:

- Λέμε τη λέξει **trigger** δηλαδή την προκαθορισμένοι λέξει την οποία έχουμε ορίσει από πριν για να ξυπνήσει η kalliope στη δική μας περίπτωση η trigger λέξη είναι Αλεξάνδρα.
- Στη συνέχεια η Αλεξάνδρα περιμένει τη φωνητική εντολή μας σε αυτό το παράδειγμα είναι "Τι ώρα είναι"
- Αν η Αλεξάνδρα καταλάβει σωστά την εντολή μας τότε θα μας απαντήσει "Η ώρα είναι 14:00"



Εικόνα 20

3.1 Πως λειτουργούσε πριν

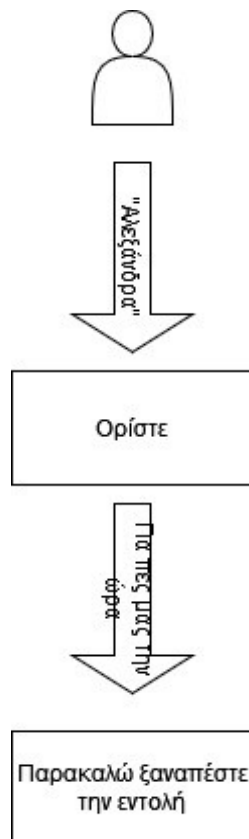
Πριν εφαρμόσουμε τη δικιά μας λύση η Αλεξάνδρα χρειαζόταν την ακριβής φράση για να μπορέσει να μας απαντήσει. Κάποια παραδείγματα για το πως λειτουργούσε πριν είναι

- Λέμε την trigger word για να ξυπνήσει η Αλεξάνδρα
- Στη συνέχεια, λέμε την εντολή που θέλουμε η οποία αν δεν την εκφράζαμε ακριβώς όπως είναι δηλωμένη στο brain δε θα μας επιστρέψει το σωστό αποτέλεσμα. Σε αυτό το παράδειγμα η εντολή είναι “Για πες μας την ώρα” η οποία δεν είναι δηλωμένη έτσι στον εγκέφαλο οπότε μας επιστέφει ότι η εντολή δεν υπάρχει ή ότι δεν την είπαμε σωστά και πρέπει να την ξανά πούμε. Σωστή διατυπώση είναι “Τι ώρα είναι”

```
Google Speech Recognition thinks you said Για πες μας την ώρα
Παρακαλώ ξαναπέστε την εντολή
Waiting for trigger detection
```

Εικόνα 21

Το διάγραμμα ροής του παραδείγματος είναι :



Εικόνα 22

Ένα ακόμα παράδειγμα είναι το εξής:

- Λέμε την trigger word για να ξυπνήσει η Αλεξάνδρα

```
Starting order signal
περιμένω την εντολή σας
Waiting for trigger detection
ορίστε !
```

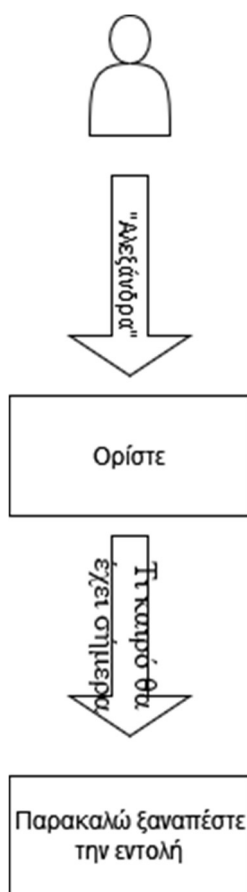
Εικόνα 23

- Λέμε την εντολή “Τι καιρό θα έχει σήμερα” η οποία μέσα στο brain είναι διατυπωμένη σαν “Τι καιρό θα κάνει” οπότε δε μας επιστρέφει τη σωστή απάντηση.

Google Speech Recognition thinks you said τι καιρό θα έχει σήμερα
 Παρακαλώ ξαναπέστε την εντολή
 Waiting for trigger detection

Εικόνα 24

Το διάγραμμα ροής του παραδείγματος είναι :



Εικόνα 25

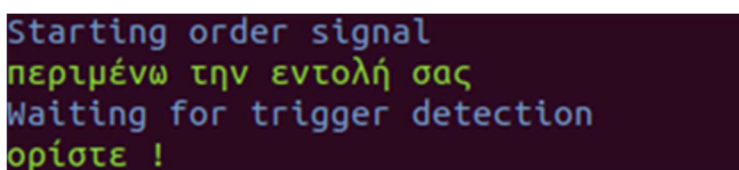
Όπως βλέπουμε, έστω και μια λέξη να διαφέρει από την προκαθορισμένη ερώτηση δε θα μας βγάλει το σωστό αποτέλεσμα.

3.2 Πως λειτουργεί τώρα

Η λύση στην οποία προτείνουμε σε αυτή την πτυχιακή είναι η έξης να επεκτείνουμε τη λειτουργικότητα του stt (μετατροπή ομιλίας σε κείμενο) προσθέτοντας έναν αλγόριθμο μηχανικής μάθησης, ώστε να μπορούμε να προσπαθήσουμε να αποτρέψουμε το πρόβλημα προτού εμφανιστεί (στην περίπτωσή μας το stt που χρησιμοποιείται είναι το google stt).

Κάποια παραδείγματα στα οποία πριν τη δικιά μας λύση επέστρεφαν λάθος απαντήσεις ενώ τώρα επιστρέφουν σωστές απαντήσεις είναι τα εξής :

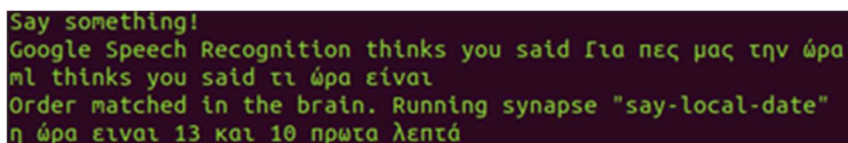
- Λέμε την trigger word για να ξυπνήσει η Αλεξάνδρα



```
Starting order signal
περιμένω την εντολή σας
Waiting for trigger detection
ορίστε !
```

Εικόνα 26

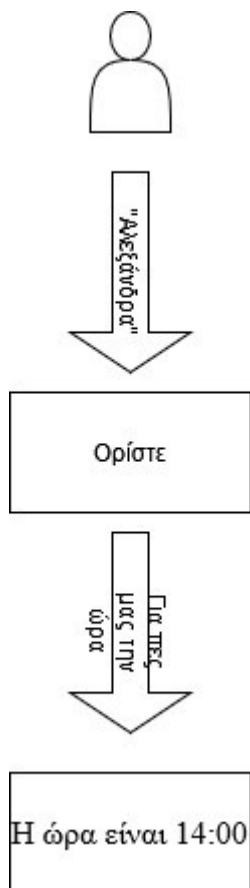
- Στη συνέχεια λέμε την εντολή που θέλουμε σε αυτό το παράδειγμα είναι “Για πες μας την ώρα”



```
Say something!
Google Speech Recognition thinks you said Για πες μας την ώρα
nl thinks you said τι ώρα είναι
Order matched in the brain. Running synapse "say-local-date"
η ώρα είναι 13 και 10 πρώτα λεπτά
```

Εικόνα 27

Το διάγραμμα ροής του συγκεκριμένου ερωτήματος είναι:



Εικόνα 28

Ένα ακόμα παράδειγμα που είδαμε πριν ότι δε μας έφερε σωστά αποτελέσματα είναι:

- Λέμε την trigger word για να ξυπνήσει η Αλεξάνδρα

```
Starting order signal  
περιμένω την εντολή σας  
Waiting for trigger detection  
ορίστε !
```

Εικόνα 29

- Λέμε την εντολή “Τι καιρό θα έχει σήμερα”

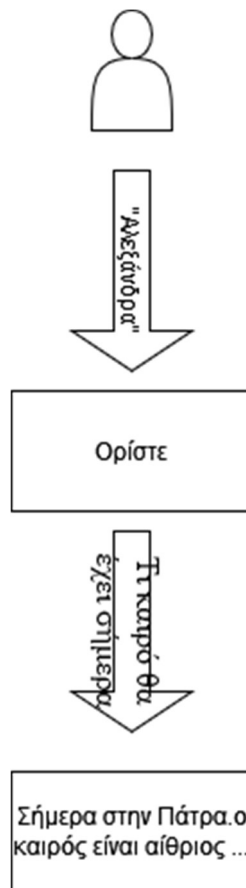
```

Say something!
Google Speech Recognition thinks you said τι καιρό θα έχει σήμερα
ml thinks you said Τι καιρό θα κάνει
Order matched in the brain. Running synapse "get-the-weather"
Σήμερα στην Patra ο καιρός είναι. αίθριος καιρός. με θερμοκρασία από 25.38 βαθμο
ύς. και αύριο ο καιρός θα είναι βροχοπτώσεις μέτριας έντασης. με θερμοκρασία απ
ό 22.61 βαθμούς

```

Εικόνα 30

Το διάγραμμα ροής του παραδείγματος είναι το εξής :



Εικόνα 31

Όπως βλέπουμε μέσω της λύσης που προτείνουμε μπορούμε να αποτρέψουμε το πρόβλημα που δημιουργούταν αν δε λέγαμε ακριβώς την ίδια εντολή.

4. Τεχνητή Νοημοσύνη (Artificial Intelligence)

Η Τεχνητή Νοημοσύνη είναι ο τομέας της επιστήμης που επιχειρεί να κατανοήσει τη λειτουργία της ανθρώπινης σκέψης και συμπεριφοράς, αλλά και να κατασκευάσει νοήμοντες οντότητες, συνδυάζοντας μία τεράστια ποικιλία επιμέρους πεδίων, τα οποία καλύπτουν ένα φάσμα που ξεκινά από γενικούς τομείς, όπως η μάθηση και η αντίληψη, και φτάνει σε συγκεκριμένες εργασίες, όπως η απόδειξη μαθηματικών θεωρημάτων και η διάγνωση

ασθενειών. Συστηματοποιεί και αυτοματοποιεί τις διανοητικές εργασίες και γι' αυτό μπορεί να έχει εφαρμογή σε οποιαδήποτε σφαίρα της ανθρώπινης διανοητικής δραστηριότητας (Russel & Norvig, 2004).

Η Τεχνητή Νοημοσύνη εστιάζεται στην ικανότητα του υπολογιστή να χειρίζεται μη αριθμητικά σύμβολα, να εξάγει συμπεράσματα και να συνάγει νέα δεδομένα από σύνολα γνωστών γεγονότων. Ένας ενδιαφέρον ορισμός της Τεχνητής Νοημοσύνης είναι: «η μελέτη των ιδεών που επιτρέπουν στους υπολογιστές να είναι ευφυείς». Δηλαδή, συστημάτων με χαρακτηριστικά τα οποία σχετίζονται με την ευφυΐα στην ανθρώπινη συμπεριφορά (μάθηση, κατανόηση φυσικής γλώσσας, αναγνώριση αντικειμένων κτλ.).

Η Τεχνητή Νοημοσύνη χωρίζεται σε δύο κατηγορίες :

Συμβολική : Η Συμβολική Τεχνητή Νοημοσύνη προσομοιώνει τον τρόπο που σκέφτονται τα άτομα, χρησιμοποιώντας ως δοκιμές μονάδες τα σύμβολα. Σε αυτήν την κατηγορία ανήκουν οι εφαρμογές της Τεχνητής Νοημοσύνης που χρησιμοποιούν αναπαράσταση γνώσης με λογική, κανόνες, πλαίσια κτλ.

Μη Συμβολική: Η μη Συμβολική Τεχνητή Νοημοσύνη προσομοιώνει τις βιολογικές διεργασίες, δηλαδή, η εξέλιξη των ειδών ή η λειτουργία του εγκεφάλου όπως τα νευρωνικά δίκτυα, οι γενετικοί αλγόριθμοι κλπ.

Κατά καιρούς έχουν διατυπωθεί διάφοροι ορισμοί της τεχνητής νοημοσύνης, από τους οποίους πολλοί επικεντρώνονται στην διαδικασία σκέψης και συλλογισμού και άλλοι στην συμπεριφορά.

- ❖ Σκέφτονται όπως οι άνθρωποι (Μηχανισμός Γνωστική επιστήμη).
- ❖ Ενεργούν όπως οι άνθρωποι (Συμπεριφορά, TuringTest)
- ❖ Σκέφτονται λογικά (Μηχανισμός, Νόμοι ορθής σκέψης)
- ❖ Αντιδρούν λογικά (Μηχανισμός, Νόμοι ορθής σκέψης)

Με την πάροδο του χρόνου και την εξέλιξη της τεχνολογίας επίπονες έρευνες επανάφεραν την χρήση των υπολογιστών για διεργασίες που απαιτούν ανθρώπινη ευφυΐα. Η πιο κοινή απάντηση στο χώρο της αγοράς για την χρήση της ΤΝ είναι η εφαρμογή των Έμπειρων Συστημάτων (Expert Systems). Τα έμπειρα συστήματα αναφέρονται και σαν Συστήματα Γνώσης (Knowledge Based Systems). Μερικοί θεωρούν ότι τα έμπειρα συστήματα δεν πρέπει να κατηγοριοποιούνται στην περιοχή της ΤΝ. Στην περιοχή αυτή, εκτός των έμπειρων συστημάτων, περιλαμβάνονται και οι παρακάτω κλάδοι τεχνολογιών, όπως:

- ❖ Νευρωνικά δίκτυα (neural networks) είναι υψηλά απλοποιημένο μοντέλο του ανθρώπινου νευρικού συστήματος, το οποίο επιδεικνύει ικανότητες, όπως εκμάθηση, γενίκευση και υπεξάιρεση. +
- ❖ Συστήματα Προοπτικής (Perspective systems) είναι συνηθισμένες εικόνες και ακουστικά σήματα που καθοδηγούν τους υπολογιστές με άλλες συσκευές. ^

- ❖ Μάθηση (learning) εμπεριέχει όλες τις ενέργειες οι οποίες διευκολύνουν έναν υπολογιστή ή μια συσκευή να αποκτήσει επιπλέον γνώση σε σχέση με αυτή που ήδη έχει στην μνήμη του από τους κατασκευαστές ή προγραμματιστές. Η μάθηση μηχανής είναι ιδιαίτερα χρήσιμη για αναγνώριση σύνθετων προτύπων και σχέσεων δεδομένων.
- ❖ Ρομποτική (robotics) αποτελείται από συσκευές οι οποίες ελέγχονται από υπολογιστή οι οποίες μιμούνται τις κινητήριες ενέργειες του ανθρώπου. Η ρομποτική χρησιμοποιείται όλο και περισσότερο στη βιομηχανία, σε εργασίες ρουτίνας, μηχανικής συναρμολόγησης προϊόντων και έχει επιτύχει μεγάλη αύξηση της παραγωγικότητας με παράλληλη μείωση του κόστους. + Υλικό της Τεχνητής Νοημοσύνης (AL Hardware) περιέχει τις φυσικές συσκευές οι οποίες στοχεύουν στην εφαρμογή της ΤΝ.
- ❖ Επεξεργασία Φυσικής Γλώσσας (Natural Language Processing) επιτρέπει στους χρήστες να επικοινωνούν με τον υπολογιστή χρησιμοποιώντας γλώσσες όπως Ελληνικά, Ισπανικά, Γερμανικά, Αγγλικά, Γαλλικά, Κινέζικα κ.λ.π.
- ❖ Fuzzy Logic είναι μια τεχνική βασισμένη στην τεχνολογιών λογικών κανόνων. Η θεωρία αυτή αναφέρει ότι οι διαχειριστικές αποφάσεις πρέπει να βασίζονται μερικώς σε πραγματικά δεδομένα, αντιλήψεις ή καταστάσεις οι οποίες μπορεί να μην είναι πάντα σωστές ή λάθος

4.1 Μηχανική Μάθηση (Machine Learning)

Η Μάθηση (Learning) είναι μία από τις θεμελιώδεις ιδιότητες της νοήμονος συμπεριφοράς του ανθρώπου. Παρά τις μελέτες και τις έρευνες επί χρόνια από τους επιστήμονες του πεδίου της Γνωστικής Ψυχολογίας και τους φιλοσόφους, η έννοια της μάθησης δεν έχει γίνει πλήρως κατανοητή. Πώς, λοιπόν, θα μπορούσαν οι επιστήμονες του χώρου της ΤΝ να δημιουργήσουν υπολογιστικά συστήματα ικανά να μάθουν, να επιτύχουν, δηλαδή, τη λεγόμενη Μηχανική Μάθηση (Machine Learning).

Αυτή μπορεί να οριστεί ως: το φαινόμενο κατά το οποίο ένα σύστημα βελτιώνει την απόδοσή του κατά την εκτέλεση μιας συγκεκριμένης εργασίας, χωρίς να υπάρχει ανάγκη να προγραμματιστεί εκ νέου.

Βάσει του ορισμού αυτού, η Μηχανική Μάθηση έχει ως σκοπό τη δημιουργία μηχανών ικανών να μαθαίνουν, να βελτιώνουν, δηλαδή, την απόδοσή τους σε κάποιους τομείς μέσω της αξιοποίησης προηγούμενης γνώσης και εμπειρίας.

Ένας σχετικός γενικός ορισμός Μηχανικής Μάθησης δίνεται από τον Mitchell (1997):

«Ένα πρόγραμμα υπολογιστή λέμε ότι μαθαίνει από την εμπειρία E ως προς κάποια κλάση εργασιών T και μέτρο απόδοσης P , αν η απόδοσή του σε εργασίες από το T , όπως μετριέται από το P , βελτιώνεται μέσω της εμπειρίας E .»

Στην Επαγωγική Μάθηση (Inductive Learning), με τη διαδικασία της επαγωγής (induction) ο άνθρωπος μαθαίνει κατανοώντας το περιβάλλον του μέσω παρατηρήσεων και δημιουργεί μια απλοποιημένη (αφαιρετική) εκδοχή του που ονομάζεται νοητικό μοντέλο (mental model). Επιπλέον, ο άνθρωπος έχει τη δυνατότητα να οργανώνει και να συσχετίζει τις εμπειρίες και τις

παρατηρήσεις του δημιουργώντας νέες δομές που ονομάζονται νοητικά πρότυπα (mental patterns), με αξιοποίηση και του επαγωγικού και του απαγωγικού συλλογισμού.

Στη δημιουργία νέων προτύπων από παλαιά βασίζονται οι τρόποι μάθησης που εξαρτώνται σε μεγαλύτερο ή μικρότερο βαθμό από την προ υπάρχουσα γνώση για ένα πρόβλημα, όπως είναι η μάθηση από επεξηγήσεις και η μάθηση από περιπτώσεις.

Σε σχέση με την ανθρώπινη ικανότητα προς μάθηση, οι φιλόσοφοι θέτουν το ερώτημα: «Πώς μπορεί ένας επαγωγικός συλλογισμός που οδηγεί στη μάθηση να αξιολογηθεί ως προς την ορθότητά του;». Αντίστοιχα, οι ψυχολόγοι ρωτούν: «Πώς αποθηκεύει ο εγκέφαλος τα αποτελέσματα της διαδικασίας της μάθησης, δηλαδή τα νοητικά μοντέλα και τα πρότυπα;».

Στο χώρο της TN απλώς ρωτούν: «Πώς μπορεί μία μηχανή να δημιουργήσει νέα μοντέλα και πρότυπα μάθησης από συγκεκριμένα παραδείγματα και πόσο αξιόπιστα είναι αυτά τα μοντέλα και πρότυπα στην πράξη;». Με βάση τα παραπάνω, μπορεί να δοθεί ο ακόλουθος εναλλακτικός ορισμός για τη Μηχανική Μάθηση: Μηχανική Μάθηση ονομάζεται η ικανότητα ενός υπολογιστικού συστήματος να δημιουργεί μοντέλα ή πρότυπα από ένα σύνολο δεδομένων.

Ως κλάδος της TN, η Μηχανική Μάθηση ασχολείται με τη μελέτη αλγορίθμων που βελτιώνουν τη συμπεριφορά τους σε κάποια εργασία που τους έχει ανατεθεί χρησιμοποιώντας την εμπειρία τους. Όσον αφορά τη σχεδίαση των συστημάτων Μηχανικής Μάθησης, για τα συστήματα που ανήκουν στη συμβολική TN, η δυνατότητα μάθησης προσδιορίζεται ως η ικανότητα πρόσκτησης επιπλέον γνώσης, που επιφέρει μεταβολές στην υπάρχουσα καταχωρημένη γνώση είτε αλλάζοντας χαρακτηριστικά της είτε με αυξομείωσή της. Στην περίπτωση των συστημάτων TN που ανήκουν στη Μη Συμβολική TN (όπως η περίπτωση των Τεχνητών Νευρωνικών Δικτύων), ως μάθηση προσδιορίζεται η δυνατότητα που διαθέτουν τα συστήματα στο να μετασχηματίζουν την εσωτερική τους δομή, παρά στο να μεταβάλλουν κατάλληλα τη γνώση που έχει καταχωρηθεί μέσα σε αυτά κατά το σχεδιασμό τους.

Αν και απέχουμε πάρα πολύ από τη δημιουργία μηχανών που μαθαίνουν τόσο καλά όσο ο άνθρωπος, για συγκεκριμένες περιοχές μάθησης έχουν αναπτυχθεί αλγόριθμοι οι οποίοι έχουν επιτρέψει την εμφάνιση σύγχρονων εμπορικών εφαρμογών με σημαντική επιτυχία. Επιπλέον, τα αποτελέσματα από τις εφαρμογές της TN αρχίζουν ήδη να είναι ορατά και να δίνουν απαντήσεις σε αναπάντητα, έως τώρα, ερωτήματα των άλλων κλάδων που διερευνούν την ικανότητα του ανθρώπου να μαθαίνει.

Ο τομέας της Μηχανικής Μάθησης αναπτύσσει, επίσης, επιτυχώς την Εξελικτική Μάθηση (Evolutionary Learning), η οποία μιμείται διαδικασίες φυσικής αναπαραγωγικής σε έμβια όντα. Χρησιμοποιείται κυρίως σε προβλήματα βελτιστοποίησης. Στην Εξελικτική Μάθηση κυριαρχούν οι γενετικοί αλγόριθμοι που θα παρουσιαστούν στο τέλος του κεφαλαίου.

Εκτός της ίδιας της TN, μεταξύ των επιστημονικών κλάδων που επωφελούνται από τα επιτεύγματα στον τομέα της Μηχανικής Μάθησης συγκαταλέγονται οι:

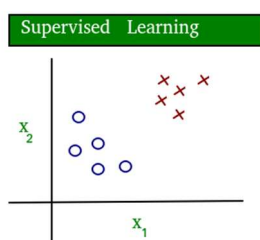
- Εξόρυξη Δεδομένων,
- Πιθανότητες και Στατιστική,
- Θεωρία της Πληροφορίας,
- Αριθμητική Βελτιστοποίηση,

- Θεωρία της Πολυπλοκότητας,
- Θεωρία Ελέγχου (προσαρμοστική),
- Ψυχολογία (εξελικτική, γνωστική),
- Νευροβιολογία και Γλωσσολογία.

4.2 Κατηγορίες μηχανικής μάθησης

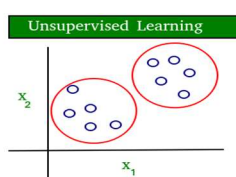
Ο τομέας της Μηχανικής Μάθησης αναπτύσσει τρεις τρόπους μάθησης, ανάλογους με τους τρόπους με τους οποίους μαθαίνει ο άνθρωπος: επιβλεπόμενη μάθηση, μη επιβλεπόμενη μάθηση και ενισχυτική μάθηση.

Στην **Επιβλεπόμενη Μάθηση (Supervised Learning)**, γνωρίζουμε τα δεδομένα μας καθώς και τις σωστές απαντήσεις τους. Με βάση αυτά τα στοιχεία, ο αλγόριθμος γενικεύει την πρόβλεψή του σε όλες τις δυνατές εισόδους.



Εικόνα 32

Στην **Μη επιβλεπόμενη Μάθηση (Unsupervised Learning)**, δεν έχουμε τις σωστές απαντήσεις στα δεδομένα μας, συνεπώς ο αλγόριθμος προσπαθεί να βρει συσχετίσεις και να γενικεύσει (κατηγοριοποιήσει) τα δεδομένα εισόδου.



Εικόνα 33

Και τέλος, η **Ενισχυτική Μάθηση (Reinforcement Learning)**, όπου ο αλγόριθμος μαθαίνει μια στρατηγική ενεργειών μέσα από άμεση αλληλεπίδραση με το περιβάλλον.

Χρησιμοποιείται κυρίως σε προβλήματα Σχεδιασμού (Planning), όπως για παράδειγμα ο έλεγχος κίνησης ρομπότ και η βελτιστοποίηση εργασιών σε εργοστασιακούς χώρους.

Υπάρχουν και άλλες κατηγορίες μάθησης, όμως εμείς θα ασχοληθούμε πιο συγκεκριμένα με την περίπτωση της **επιβλεπόμενης μάθησης**.

4.3 Επιβλεπόμενη Μάθηση

Το χαρακτηριστικό αυτής είναι ότι μας δίνονται τα δεδομένα εισόδου, συνήθως με την μορφή διανυσμάτων, καθώς και το σωστό αποτέλεσμα στην έξοδο του αλγορίθμου. Στόχος μας είναι η εκπαίδευση του αλγορίθμου ώστε να μάθει από τα δεδομένα που γνωρίζουμε τις απαντήσεις τους και κατόπιν να γενικεύσουμε, έτσι ώστε όταν παρουσιάσουμε στον αλγόριθμο νέα δεδομένα με άγνωστες απαντήσεις, και αυτός να μπορέσει να βρει τις σωστές απαντήσεις με βάση την εμπειρία που απέκτησε από τα γνωστά δεδομένα. Ουσιαστικά, στην επιβλεπόμενη μάθηση, το σύστημα προσεγγίζει μια συνάρτηση στόχο, η οποία περιγράφει το φαινόμενο που εξετάζουμε.

Τα δεδομένα εκπαίδευσης (training data set) συνήθως συμβολίζονται σαν σύνολα (x_i, t_i) , όπου x_i είναι τα διανύσματα των δεδομένων εισόδου ενώ t_i είναι το αποτέλεσμα κάθε εισόδου i .

Το σύνολο των διανυσμάτων εισόδου αποτελούν το πεδίο ορισμού της συνάρτησης που επιχειρούμε να προσεγγίσουμε και ονομάζονται στιγμιότυπα (instances).

Κάθε στιγμιότυπο (διάνυσμα) μπορεί να έχει πολλές διαστάσεις, τις οποίες ονομάζουμε χαρακτηριστικά (attributes).

Ανάλογα με τα δεδομένα εξόδου που έχουμε, χαρακτηρίζουμε το σύνολό μας ως συνεχές ή διακριτό.

Συνεχές είναι στην περίπτωση όπου τα t_i μπορούν να πάρουν οποιαδήποτε τιμή, για παράδειγμα δεδομένα χρόνου, απόστασης κλπ.

Αντίθετα, στα διακριτά δεδομένα τα t_i μπορούν να πάρουν συγκεκριμένες τιμές μέσα από κάποιο πεπερασμένο σύνολο τιμών, για παράδειγμα είναι (true, false), (0,1) κα.

Οι διακριτές τιμές που μπορούν να πάρουν μπορεί να είναι πολλές (Πόλεις μιας χώρας, ονόματα φυτών κλπ.) ή λίγες (το φύλο ενός προσώπου), είναι όμως πάντα συγκεκριμένες και αριθμήσιμες.

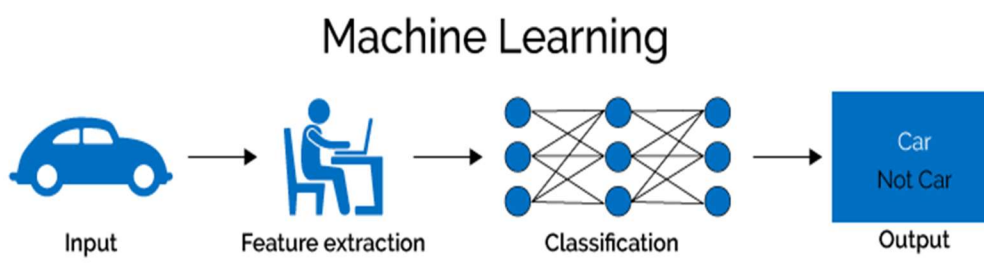
Με βάση την προηγούμενη κατηγοριοποίηση των δεδομένων εξόδου, έχουμε και τον χαρακτηρισμό των προβλημάτων που μελετούμε σε προβλήματα πρόβλεψης και κατηγοριοποίησης

Η Επιβλεπόμενη Μάθηση μπορεί να χωριστεί σε 2 κατηγορίες, δηλαδή ταξινόμηση & παλινδρόμηση. Εμείς θα ασχοληθούμε με την ταξινόμηση.

Εναλλακτική ορολογία:

- **Στιγμιότυπα (instances)** όπου επίσης αναφέρονται ως παραδείγματα ή δείγματα (examples)
- **Γνωρίσματα (attributes)** όπου επίσης γνωστά ως χαρακτηριστικά (features), προσόντα (qualifications), μεταβλητές (variables), ανεξάρτητες μεταβλητές (independent variables), μεταβλητές εισόδου (input variables), συμεταβλητές (covariates)

- **Μεταβλητή στόχου (goal variable)** ή γνωστή ως κατηγορημα στόχου (goal predicate), εξαρτημένη μεταβλητή (dependent variable), τάξη κατηγοριοποίησης (classification class)
- **Συνάρτηση λάθους (error function)** ,γνωστή ως αντικειμενική συνάρτηση (objective function), συνάρτηση απώλειας (loss function).



Εικόνα 34

4.4 Μη Επιβλεπόμενη μάθηση

Η μη επιβλεπόμενη μάθηση είναι όπου δέχετε μόνο δεδομένα εισόδου (X) και καμία αντίστοιχη μεταβλητή εξόδου.

Ο στόχος για την μη επιβλεπόμενη μάθηση είναι να μοντελοποιήσει την υποκείμενη δομή ή τη διανομή στα δεδομένα προκειμένου να μάθει περισσότερα για αυτά.

Σε αντίθεση με την επιβλεπόμενη μάθηση που αναφέραμε παραπάνω δεν υπάρχουν σωστές απαντήσεις και δεν υπάρχει δάσκαλος. Οι αλγόριθμοι αφήνονται στις δικές τους διατάξεις για να ανακαλύψουν και να παρουσιάσουν την ενδιαφέρουσα δομή στα δεδομένα.

Μερικά δημοφιλή παραδείγματα αλγορίθμων μάθησης χωρίς επίβλεψη είναι:

- k-means για προβλήματα ομαδοποίησης.
- Αλγόριθμος Apriori για προβλήματα μάθησης κατά κανόνα σύνδεσης.

4.5 Αλγόριθμοι ταξινόμησης (Classification Algorithms)

Στην κατηγορία αυτή έχουμε διακριτά δεδομένα, αντί για συνεχή. Οι τιμές εξόδου, t_i , αντιστοιχούν σε κατηγορίες στις οποίες κατατάσσονται τα διανύσματα εισόδου.

Συνεπώς, παρουσιάζοντας νέα δεδομένα, δεν προσπαθούμε να προσεγγίσουμε την τιμή μιας συνάρτησης, αλλά να κατατάξουμε την εισοδό μας σε κάποια από τις κατηγορίες που γνωρίζουμε. Οι κατηγορίες αυτές συνήθως ονομάζονται κλάσεις και μπορεί να έχουν αριθμητικές τιμές, να είναι σύμβολα κλπ. Γενικά ανάλογα με την φύση του προβλήματος που μελετούμε καθορίζονται και οι κλάσεις στις οποίες ομαδοποιούνται τα δεδομένα μας. Οι μεθοδολογίες που εφαρμόζονται στα προβλήματα κατηγοριοποίησης διαφέρουν πολύ μεταξύ τους σε ότι αφορά τον τρόπο με τον οποίο προσπαθούν να ομαδοποιήσουν τα στοιχεία που

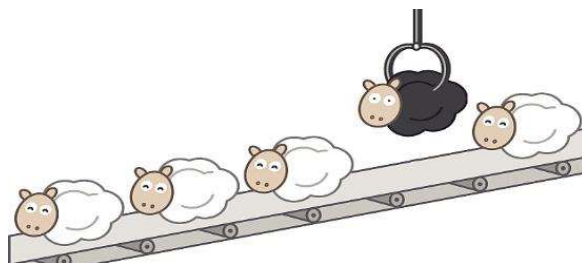
τους παρέχονται. Στην γενική τους όμως μορφή, εκείνο που προσπαθούν να κάνουν είναι να υπολογίσουν τα όρια μεταξύ των κατηγοριών - κλάσεων (decision boundaries).

Για παράδειγμα έστω, ότι θέλουμε να κάνουμε ανίχνευση ανεπιθύμητων μηνυμάτων σε παρόχους υπηρεσιών ηλεκτρονικού ταχυδρομείου. Πρόκειται για μία ταξινόμηση δυαδική αφού έχουμε μόνο 2 κλάσεις. Την «spam» και «όχι spam».

Ένας ταξινομητής χρησιμοποιεί ορισμένα στοιχεία εκπαίδευσης για να κατανοήσει τον τρόπο με τον οποίο οι δεδομένες μεταβλητές εισόδου σχετίζονται με την κλάση.

Σε αυτή την περίπτωση, τα μηνύματα ηλεκτρονικού ταχυδρομείου ανεπιθύμητης αλληλογραφίας spam και μηνύματα τα οποία δεν είναι spam πρέπει να χρησιμοποιούνται ως δεδομένα εκπαίδευσης.

Όταν ο ταξινομητής εκπαιδευτεί με ακρίβεια, μπορεί να χρησιμοποιηθεί για την ανίχνευση ενός μηνύματος το οποίο είναι άγνωστο, δηλαδή η κλάση του είναι κενή.



Εικόνα 35

4.5.1 Κριτήρια αξιολόγησης μεθόδων κατηγοριοποίησης

Η έρευνα σχετικά με την κατηγοριοποίηση έχει αποδώσει πλούσιους καρπούς και σήμερα υπάρχουν διαθέσιμες αρκετές και πολύ διαφορετικές μέθοδοι κατηγοριοποίησης. Ορισμένες από αυτές, όπως πχ τα Νευρωνικά - 207 - Δίκτυα, θεωρούνται ιδιαίτερα ικανές να προβλέπουν την κλάση άγνωστων παρατηρήσεων. Οι μέθοδοι αυτές μπορούν να θεωρηθούν «καλύτερες» από άλλες, όμως η ακρίβεια δεν είναι το μοναδικό κριτήριο αξιολόγησης των μεθόδων κατηγοριοποίησης. Αναλυτικότερα, οι μέθοδοι κατηγοριοποίησης μπορούν να αξιολογηθούν με βάση τα παρακάτω κριτήρια:

Ακρίβεια πρόβλεψης (accuracy). Είναι η ικανότητα των μοντέλων να προβλέπουν την κλάση άγνωστων παρατηρήσεων. Προφανώς πρόκειται για ένα πολύ σημαντικό κριτήριο και μεγάλο μέρος της έρευνας προσανατολίζεται στην ανακάλυψη μεθόδων υψηλών επιδόσεων.

Ταχύτητα (speed). Σχετίζεται με την πολυπλοκότητα της μεθόδου και το υπολογιστικό κόστος που αυτή συνεπάγεται. Η εκτέλεση περίπλοκων αλγορίθμων, οι οποίοι απαιτούν εκτεταμένους υπολογισμούς, προκαλούν καθυστερήσεις. Καθυστερήσεις μπορεί να υπάρχουν στη διαδικασία κατασκευής, αλλά και στη χρήση των μοντέλων, στην εφαρμογή τους δηλαδή για την κατηγοριοποίηση μιας νέας παρατήρησης. Ορισμένες μέθοδοι, όπως τα Δένδρα Αποφάσεων, διαθέτουν γρήγορους αλγορίθμους και ο χρόνος κατασκευής των μοντέλων είναι μικρός. Άλλες μέθοδοι, όπως τα Νευρωνικά Δίκτυα, χρειάζονται πολύ περισσότερο

χρόνο για την εκπαίδευση των μοντέλων. Κατά κανόνα ο χρόνος χρήσης των μοντέλων είναι πολύ μικρός. Ωστόσο, υπάρχουν μέθοδοι, όπως οι k-Πλησιέστεροι Γείτονες, οι οποίες δεν εκπαιδεύουν κάποιο μοντέλο, όμως ο χρόνος για την κατηγοριοποίηση νέων παρατηρήσεων είναι μεγάλος

Ερμηνευση (interpretability). Είναι η ικανότητα της μεθόδου να παράγει μοντέλα, τα οποία είναι κατανοητά από τον άνθρωπο. Για παράδειγμα, στα Δένδρα Αποφάσεων ο τρόπος λήψης της απόφασης κατηγοριοποίησης είναι απολύτως κατανοητός και το μοντέλο μπορεί εύκολα να μετατραπεί σε ένα σύνολο κανόνων της μορφής EAN-TOTE. Αντιθέτως, τα μοντέλα άλλων μεθόδων, όπως τα Νευρωνικά Δίκτυα και οι Μηχανές Διανυσμάτων Υποστήριξης, λειτουργούν ως «μαύρα κουτιά». Στα μοντέλα αυτά παρέχονται οι τιμές των μεταβλητών εισόδου και υπολογίζεται η απόφαση κατηγοριοποίησης στην έξοδο. Ο τρόπος λήψης της απόφασης όμως δεν είναι κατανοητός στον άνθρωπο. Η ερμηνευση είναι μια σημαντική ιδιότητα των μεθόδων κατηγοριοποίησης. Σε πολλές περιπτώσεις οι χρήστες των μοντέλων επιθυμούν να γνωρίζουν τον τρόπο λήψης της απόφασης, ώστε να είναι πιο σίγουροι για το αποτέλεσμα. Επίσης, στο μοντέλο καταγράφονται σχέσεις μεταξύ των δεδομένων. Ορισμένες από τις σχέσεις αυτές μπορεί να είναι νέες και άγνωστες. Αν το μοντέλο είναι ερμηνεύσιμο θα αποκαλυφθούν οι νέες σχέσεις και η μέθοδος κατηγοριοποίησης θα χρησιμοποιηθεί ως εργαλείο ανάλυσης, ικανό να προσφέρει καινοτόμα γνώση.

Επεκτασιμότητα (scalability). Αναφέρεται στην ικανότητα των μεθόδων να χειριστούν πολύ μεγάλα σύνολα δεδομένων. Η Μηχανική Μάθηση και η Στατιστική προσφέρουν μεθόδους κατηγοριοποίησης. Ωστόσο, η εφαρμογή αυτών των μεθόδων για την επεξεργασία δεδομένων μεγάλου όγκου δεν είναι πάντα εύκολη. Σε αρκετές περιπτώσεις η υπολογιστική πολυπλοκότητα των μεθόδων είναι συνάρτηση του πλήθους των παρατηρήσεων και μάλιστα με σχέση περισσότερο από γραμμική. Επίσης, οι περισσότερες μέθοδοι απαιτούν την εγκατάσταση του συνόλου εκπαίδευσης στην κύρια μνήμη του υπολογιστή. Τα ζητήματα αυτά θέτουν όρια στη δυνατότητα εφαρμογής των μεθόδων.

Ανθεκτικότητα (robustness). Αναφέρεται στην ικανότητα των μεθόδων να πραγματοποιήσουν ορθές προβλέψεις, όταν τα δεδομένα χαρακτηρίζονται από προβλήματα, όπως ο θόρυβος και οι χαμένες τιμές.

4.5.2 Ταξινόμηση δέντρων αποφάσεων - Decision Tree Classification

Τα Δένδρα Απόφασης-ΔΑ (Decision Trees) είναι ο γνωστότερος αλγόριθμος επιβλεπόμενης Επαγωγικής Μάθησης και έχει εφαρμοστεί με επιτυχία σε πολλούς τομείς όπου απαιτείται ταξινόμηση: ενδεικτικά, στην αναγνώριση προσώπων σε εικόνες, στην ιατρική για διάγνωση περιστατικών, για προβλέψεις απαραίτητες στη διαφήμιση, για προώθηση προϊόντων και, γενικότερα, για εξόρυξη γνώσης. Τα δέντρα ταξινόμησης χρησιμοποιούνται για να προβλέψουν, με κάποιο βαθμό ακρίβειας, την τιμή της μεταβλητής που μοντελοποιούν με βάση τις τιμές των θεωρούμενων ανεξάρτητων χαρακτηριστικών. Ένα δέντρο απόφασης αναπαριστά μια διαδικασία λήψης απόφασης, όπου για κάθε πιθανό σημείο έχουμε έναν κόμβο, ενώ για κάθε πιθανή επιλογή που μπορεί να γίνει σε ένα σημείο απόφασης αναπαριστάται με ένα «κόμβο-παιδί». Κάθε κόμβος ορίζει μια συνθήκη ελέγχου της τιμής κάποιου χαρακτηριστικού των περιπτώσεων. Κάθε κλαδί που φεύγει από ένα κόμβο αντιστοιχεί σε μια διαφορετική διακριτή τιμή του χαρακτηριστικού που σχετίζεται με τον

κόμβο. Στα κλαδιά καταλήγουν οι τελικοί κόμβοι που ανήκουν σε ένα μόνο σύνολο, όπου είναι οι τελικές αποφάσεις. Για να διασπάσουμε ένα κόμβο (parent) με N εγγραφές σε K παιδιά u_i και ο αριθμός εγγραφών είναι $N(u_i)$ με $\sum_{i=1}^K N(u_i) = N$ Για να διαλέξουμε τη διάσπαση με το μεγαλύτερο κέρδος, υπολογίζουμε το μέγιστο Δ από την επόμενη σχέση

$$\Delta = I(\text{parent}) - \sum_{i=1}^K \frac{N(u_i)}{N} i(u_i)$$

Μέτρα μη-Καθαρότητας

Στη κατασκευή ενός δέντρου παρουσιάζονται τρία μέτρα μη-καθαρότητας: η εντροπία, το ευρετήριο Gini και τέλος το λάθος ταξινόμησης. Η εντροπία είναι ο βαθμός ή η ποσότητα της αβεβαιότητας στην τυχειότητα των στοιχείων ή με άλλα λόγια είναι ένα μέτρο της ακαθαρσίας. Ορίζεται από τον τύπο:

$$E(S) = - \sum_{i=1}^c p_i \log p_i$$

Όπου S είναι ένα σύνολο δεδομένων, P_i είναι το ποσοστό των παραδειγμάτων του S που ανήκουν στην κατηγορία I_k και c διαφορετικές κατηγορίες.

Η **εντροπία** υπολογίζει την ομοιογένεια ενός δείγματος. Αν το δείγμα είναι εντελώς ομοιογενές, η εντροπία είναι μηδέν και αν το δείγμα είναι εξίσου διαιρεμένο έχει μια εντροπία ενός

Το **κέρδος πληροφορίας (information gain)** μετράει τη σχετική μεταβολή της εντροπίας σε σχέση με την ανεξάρτητη ιδιότητα. Προσπαθεί να υπολογίσει τις πληροφορίες που περιέχει κάθε ιδιότητα. Η κατασκευή ενός δέντρου αποφάσεων αφορά στην εύρεση του χαρακτηριστικού που επιστρέφει το υψηλότερο κέρδος πληροφορίας (δηλαδή, οι πιο ομοιογενείς κλάδοι).

$$Gain(T, X) = Entropy(T) - Entropy(T, X)$$

Εικόνα 36

Όπου το κέρδος (T, X) είναι το κέρδος πληροφορίας με την εφαρμογή του χαρακτηριστικού X . Η εντροπία (T) είναι η Εντροπία ολόκληρης της σειράς, ενώ ο δεύτερος όρος υπολογίζει την Εντροπία μετά την εφαρμογή του χαρακτηριστικού X . Το μειονέκτημα ενός μοντέλου δέντρω αποφάσεων είναι η υπερφόρτωση καθώς προσπαθεί να προσαρμόσει το μοντέλο περνώντας βαθύτερα στο σύνολο εκπαίδευσης και μειώνοντας έτσι την ακρίβεια της δοκιμής.

Το ευρετήριο gini για κάθε κόμβο δίνετε από τον τύπο:

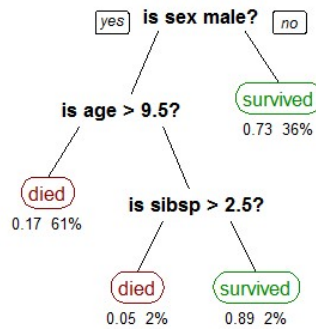
$$GINI(t) = 1 - \sum [p(j/t)]^2$$

Η ελάχιστη τιμή που μπορεί να πάρει είναι $(0,0)$, όταν όλες οι εγγραφές ανήκουν σε μια κλάση ενώ η μέγιστη τιμή που μπορεί να πάρει είναι $(1-1/c)$, όταν όλες οι εγγραφές είναι ομοιόμορφα κατανεμημένες στις κλάσεις.

Πλεονεκτήματα/ Μειονεκτήματα

Το δέντρο απόφασης είναι απλό να κατανοηθεί και να απεικονιστεί, απαιτεί λίγη προετοιμασία δεδομένων και μπορεί να χειριστεί τόσο τα αριθμητικά όσο και τα κατηγορηματικά δεδομένα.

Ένα από τα αρνητικά των δέντρων αποφάσεων είναι ότι μπορεί να δημιουργούν σύνθετα δέντρα που δεν γενικεύουν σωστά και μπορεί να γίνουν ασταθή επειδή μικρές παραλλαγές στα δεδομένα ενδέχεται να οδηγήσουν σε ένα εντελώς διαφορετικό δέντρο που παράγεται



Εικόνα 37

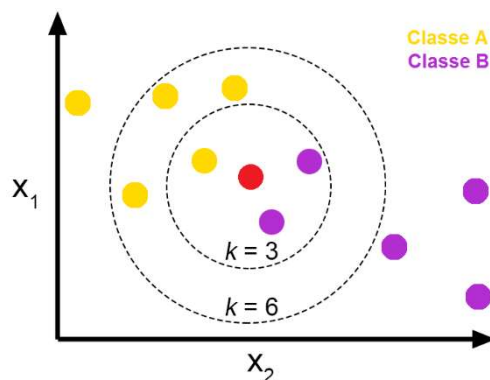
4.5.3 Algorithm K Nearest Neighbors - K Κοντινότεροι Γείτονες

Ο αλγόριθμος ταξινόμησης με βάση τους k κοντινότερους γείτονες αποτελεί μια βασική μέθοδο μάθησης βασισμένη σε δείγματα [62]. Η λογική του αλγορίθμου είναι πολύ απλή καθώς το υπό κατηγοριοποίηση πρότυπο κατατάσσεται στην κλάση που ανήκουν τα k - κοντινότερα σε αυτό πρότυπα εκπαίδευσης. Η απόσταση ορίζεται ως η Ευκλείδεια απόσταση του δείγματος προς εξέταση και όλων των υπάρχοντων δειγμάτων. Ο τύπος που δίδεται η απόσταση μεταξύ των σημείων p, q σε έναν χώρο που ανήκει στον R^n ορίζεται ως:

$$d(p, q) = \sqrt{(p_1 - q_1)^2 + (p_2 - q_2)^2 + \dots + (p_n - q_n)^2}$$

Επιλέγονται τελικώς τα k-δείγματα με τις μικρότερες αποστάσεις. Η επιλογή της μεταβλητής k εξαρτάται από το δείγμα δεδομένων προς ταξινόμηση. Η απόδοση του αλγορίθμου μειώνεται όσο αυξάνεται το k. Αυτό συμβαίνει διότι οι αποστάσεις μεταξύ των κλάσεων μειώνονται δυσχεραίνοντας το έργο του ταξινομητή. Ένα πλεονέκτημα των ταξινομητών k-NN σε σχέση με τα ΔΑ και τα νευρωνικά δίκτυα αποτελεί η σταθερότητα τους. Μια μέθοδος μάθησης χαρακτηρίζεται ως «ασταθής» εάν οι μικρές μεταβολές στα δεδομένα εκπαίδευσης προκαλούν σημαντικές μεταβολές στα αποτελέσματα του ταξινομητή. Αυτός ο αλγόριθμος επηρεάζεται σημαντικά από τα δεδομένα που λαμβάνουν πολύ μεγάλες ή μικρές τιμές αντίστοιχα, με αποτέλεσμα να μειώνεται η ακρίβεια του.

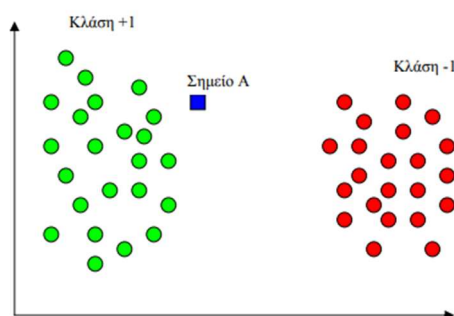
Στο επόμενο κεφάλαιο χρησιμοποιείται μια ειδική κατηγορία του kNN, ο kNN με βάρη (weighted kNN). Αυτός αναθέτει σε κάθε γείτονα ένα βάρος w, και με βάση αυτό επιλέγεται ο κοντινότερος γείτονας, καθώς και η κλάση του δείγματος. Στην εργασία αυτή τα βάρη ακολουθούν την συνάρτηση του αντίστροφου τετραγώνου. Στην περίπτωση που δεν επιλεχθεί κάποιο βάρος, η μέθοδος ονομάζεται fine kNN.



Εικόνα 38

4.5.4 Μηχανές διανυσμάτων υποστήριξης (Support Vector Machine)

Στο παράδειγμα βλέπουμε ένα σύνολο σημείων x_i στο δισδιάστατο χώρο, καθώς και την κλάση στην οποία ανήκουν (+1 ή -1). Στόχος μας είναι να εκπαιδεύσουμε τα SVM ώστε να μάθουν την σχέση σημείου – κλάσης για τα δοσμένα σημεία, έτσι ώστε, για ένα άλλο σημείο (έστω A) να μας προβλέψουν σε ποια κλάση ανήκει.

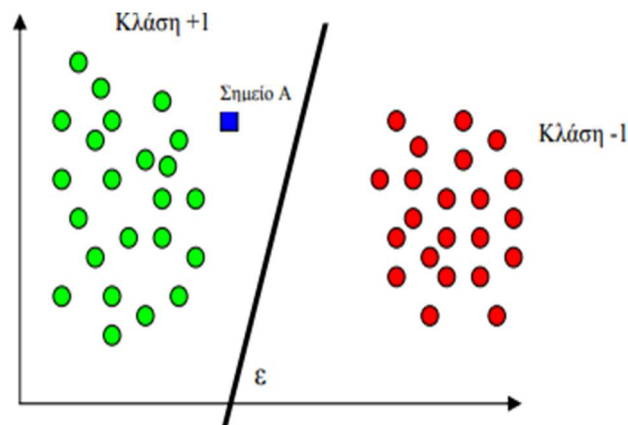


Εικόνα 39

Για να αποφασιστεί σε ποια κλάση ανήκει το νέο σημείο, θα πρέπει να βρεθεί που είναι το όριο της κάθε κλάσης, δηλαδή να βρεθεί μια γραμμή (δισδιάστατος χώρος) που να διαχωρίζει τις δύο κλάσεις.

Τα SVMs, για να πετυχαίνουν το διαχωρισμό των κλάσεων χρησιμοποιούν ευθείες. Έτσι, από τη σχετική θέση του προς κατάταξη σημείου και της διαχωριστικής ευθείας θα μπορεί να βγει το συμπέρασμα σε ποια κλάση ανήκει το σημείο.

Στο επόμενο σχήμα βλέπουμε μια διαχωριστική ευθεία (ϵ). Επειδή το σημείο A βρίσκεται στα αριστερά της ευθείας, εκεί δηλαδή που υπάρχει η κλάση +1, συμπεραίνουμε πως το σημείο A ανήκει στην κλάση +1.



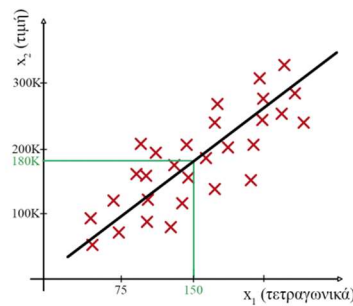
Εικόνα 40

Με βάση τα παραπάνω, στόχος μας είναι να προσδιοριστεί η βέλτιστη ευθεία διαχωρισμού των κλάσεων, για την οποία θα επιτυγχάνεται η ελαχιστοποίηση του σφάλματος κατάταξης. Δηλαδή, να καταταχθεί σωστά, στην κλάση που πραγματικά ανήκει, όσο το δυνατόν μεγαλύτερος αριθμός σημείων. Για να το πετύχουμε αυτό, μια λογική υπόθεση θα ήταν να απαιτήσουμε η διαχωριστική μας ευθεία (που καθορίζει το όριο των κλάσεων) να μην βρίσκεται κοντά στα δεδομένα σημεία των κλάσεων.

4.5.5 Γραμμική παλινδρόμηση (Linear Regression)

Η γραμμική παλινδρόμηση αποτελεί την απλούστερη μορφή παλινδρόμησης. Έχει ως στόχο την εκπαίδευση (training) μιας συνάρτησης, η οποία απεικονίζει ένα αντικείμενο σε μία πραγματική μεταβλητή.

Στην Εικόνα παρουσιάζουμε ένα απλό παράδειγμα γραμμικής παλινδρόμησης. Οι μεταβλητές είναι τα τετραγωνικά μας σπιτιού και η τιμή πώλησής του σε χιλιάδες Ευρώ. Η γραμμική παλινδρόμηση προσαρμόζει μια ευθεία στα δείγματα του συνόλου δεδομένων, τα οποία σηματοδοτούνται με κόκκινο X. Η προσαρμογή γίνεται με βάση μια συνάρτηση κόστους, την τιμή μας οποία θέλουμε να ελαχιστοποιήσουμε. Έχοντας τη βέλτιστη ευθεία, δηλαδή την ευθεία που ελαχιστοποιεί την τιμή μας συνάρτησης κόστους, μπορούμε να δώσουμε μια προσεγγιστικά καλή απάντηση σε ερωτήματα μας μορφής: «Σε τι τιμές πωλούνται σπίτια των 150 τετραγωνικών;». Δηλαδή, δοθέντων των τιμών μας μεταβλητής στόχου (στην περίπτωση μας η τιμή πώλησης) για κάθε δείγμα, επιδιώκουμε να προβλέψουμε μας τιμές μας μεταβλητής στόχου για νέα δείγματα. Στο πλαίσιο παρουσίασης μας γραμμικής παλινδρόμησης θα ορίσουμε κάποιους συμβολισμούς. Θα συμβολίσουμε με m το πλήθος των δειγμάτων του συνόλου εκπαίδευσης. Με X θα συμβολίσουμε μας μεταβλητές εισόδου, ενώ με y τη μεταβλητή στόχου



Εικόνα 41

4.5.6 Λογική παλινδρόμηση (Logistic Regression)

Η λογιστική παλινδρόμηση (Logistic regression) αποτελεί στην ουσία ένα μοντέλο ταξινόμησης των τιμών μιας μεταβλητής απόκρισης Y με βάση τη θεωρία των πιθανοτήτων. Στο μοντέλο αυτό όπου η μεταβλητή Y συνήθως έχει δυαδικό χαρακτήρα (λαμβάνει δύο τιμές) στοχεύεται η πρόβλεψη μας έκβασης μας από ένα πλήθος προβλεπτικών μεταβλητών που μπορεί να είναι ονομαστικές, τακτικές ή ποσοτικές. Η σημαντικότερη διαφοροποίηση μεταξύ λογιστικής και γραμμικής παλινδρόμησης βασίζεται στη φύση μας επιλεγμένης μεταβλητής απόκρισης, η οποία στην μεν πρώτη μπορεί να είναι κατηγορική, (τακτική ή ονομαστική, στη δε δεύτερη αποκλειστικά ποσοτική. Ενώ κατά την κλασική γραμμική παλινδρόμηση η εκτίμηση των παραμέτρων a και b γίνεται με τη μέθοδο των ελάχιστων τετραγώνων, κατά τη λογιστική παλινδρόμηση η εκτίμηση των παραμέτρων γίνεται με τη μέθοδο του λόγου πιθανοφάνειας (μέθοδος συνήθως εφαρμοζόμενη στα γενικευμένα γραμμικά υποδείγματα), δηλαδή επιλέγονται οι πιο πιθανοφανείς τιμές των παραμέτρων, προκειμένου να οδηγήσουν στα παρατηρούμενα αποτελέσματα. Ως επακόλουθο, η πρώτη παραδέχεται την ύπαρξη ομοιογένειας (ομοσκεδαστικότητας) στα υπολείμματα των αποκρίσεων ενώ στη δεύτερη αναπτύσσεται πάντα ετεροσκεδαστικότητα σε κάθε προβλεπόμενη τιμή εξαιτίας του μεταβαλλόμενου ποσοστού διακύμανσης που αναλογεί σε αυτήν.

Διακρίνονται τρεις τύποι λογιστικής παλινδρόμησης ανάλογα με την ιδιαίτερη φύση μας εξαρτημένης κατηγορικής μεταβλητής η οποία μπορεί να είναι:

- **Δίτιμη ή δυαδική ή διχοτομική (binary) ή διμερής εξαρτημένη μεταβλητή.** Συνίσταται από δύο κατηγορίες, μας π.χ. είναι οι εκβάσεις επιτυχία/αποτυχία, ΝΑΙ/ΜΑΣ, γεγονός απόν/παρόν.
- **Τακτική (ordinal) μεταβλητή.** Η εξαρτημένη μεταβλητή συνίσταται από τρεις ή περισσότερες κατηγορίες μεταξύ των οποίων ισχύει η έννοια μας ανισότητας, μας π.χ. σε μια ερώτηση μας κλίμακας διαφωνώ καθόλου, λίγο, μέτρια, αρκετά, πολύ, στην κατάταξη μας στρώματος υλικού ως λεπτού, μεσαίου, παχέος.
- **Ονομαστική (Nominal) ή πολυωνυμική (polynomial) ή πολυχοτομική (polychotomus) ή κατηγορική αδιαβάθμητη (non-ordered categorical) ή πολυμερής μεταβλητή απόκρισης.** Περιέχει τρεις ή περισσότερες κατηγορίες χωρίς κάποια φυσική διαβάθμιση, μας π.χ. ο χαρακτηρισμός μας τροφίμου ως τραγανού, μαλακού, εύθρυπτου ή του χρώματος αντικειμένων ως ερυθρού, πράσινου, κίτρινου κτλ.

4.5.7 Naive Bayes

Ο Αφελής Μπαϋεσιανός κατηγοριοποιητής αποτελεί εφαρμογή του θεωρήματος Bayes. Υποθέτουμε ότι X είναι μια παρατήρηση του συνόλου δεδομένων και H είναι η υπόθεση ότι παρατήρηση αυτή ανήκει στην κλάση C_i .

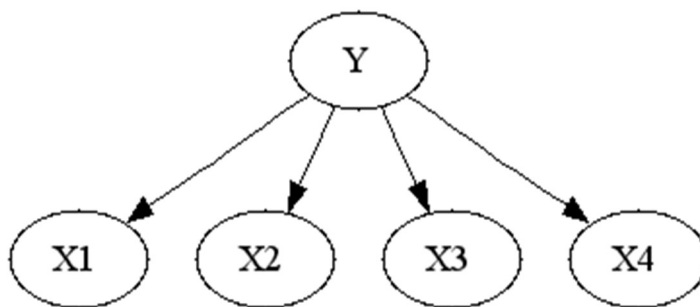
Πιο συγκεκριμένα, το X θεωρείται ως ένα άνυσμα n τιμών $X=(x_1, x_2, \dots, x_n)$. Υποθέτουμε ότι υπάρχουν m κλάσεις C_1, C_2, \dots, C_m . Σύμφωνα με το θεώρημα του Bayes, η πιθανότητα να ανήκει η παρατήρηση X στην κλάση C_i υπάρχουν m κλάσεις υπολογίζεται από την Εξίσωση 9.13. C_1, C_2, \dots, C_m . Σύμφωνα με το θεώρημα του Bayes, η πιθανότητα να ανήκει η παρατήρηση X στην κλάση C_i υπολογίζεται από την Εξίσωση .

$$p(C_i|X) = P(C_i) * \frac{P(C_i)}{P(X)}$$

Για να προβλέψει την κλάση μιας άγνωστης παρατήρησης, ο Αφελής Μπαϋεσιανός κατηγοριοποιητής υπολογίζει μας πιθανότητες για την κάθε κλάση και εκχωρεί την παρατήρηση στην κλάση με τη μεγαλύτερη πιθανότητα. Εφόσον το $P(X)$ είναι ίδιο για μας μας κλάσεις και το $P(C_i)$ μπορεί εύκολα να υπολογιστεί (ως το πλήθος των παρατηρήσεων που ανήκουν στην κλάση C_i μας το πλήθος όλων των παρατηρήσεων), το ζητούμενο είναι ο υπολογισμός του $P(X|C_i)$. Ο υπολογισμός του $P(X|C_i)$ μπορεί να αποδειχθεί ιδιαίτερα περίπλοκος εάν θεωρηθεί ότι υπάρχει σχέση εξάρτησης μεταξύ των διαστάσεων του ανύσματος X , δηλαδή μεταξύ των μεταβλητών εισόδου. Αντιθέτως, αν θεωρηθεί ότι, δοθείσης μας κλάσης, οι μεταβλητές εισόδου είναι μεταξύ μας ανεξάρτητες, τότε ο υπολογισμός του $P(X|C_i)$ απλοποιείται και δίνεται από την Εξίσωση

$$p(x|C_i) = \prod_{k=1}^n p(x_k|C_i)$$

Όπου X_k είναι η τιμή μας διάστασης k του ανύσματος X . Μας Αφελής Μπαϋεσιανός κατηγοριοποιητής με τέσσερεις ανεξάρτητες μεταβλητές παρουσιάζεται με μορφή γραφικού πιθανοτικού μοντέλου στο Σχήμα.



Εικόνα 42

Ο κατηγοριοποιητής, αφού υπολογίσει μας πιθανότητες $P(C_i|X)$ για μας κλάσεις C_i , εκχωρεί την παρατήρηση στην κλάση με τη μεγαλύτερη πιθανότητα. Εάν ισχύει η υπόθεση ότι δεδομένης μας κλάσης είναι ανεξάρτητες οι μεταβλητές εισόδου, ο Αφελής Μπαϋεσιανός

κατηγοριοποιητής επιτυγχάνει μας υψηλότερους ρυθμούς ακρίβειας. Ωστόσο, στην πράξη μας περισσότερες φορές η υπόθεση αυτή δεν ισχύει.

4.5.8 Random Forest

Ο αλγόριθμος μας αναπτύχθηκε από μας Leo Breiman και Adele Cutler το 2001. Είναι από μας δημοφιλέστερους αλγόριθμους στην κατηγορία του, κυρίως για την ταχύτητα αλλά και την ακρίβεια που προσφέρει.

Σύμφωνα με μας δημιουργούς του, προσφέρει την καλύτερη ακρίβεια μεταξύ των υπαρχόντων αλγορίθμων.

Η ταχύτητά του είναι πολύ καλή ακόμα και σε πολύ μεγάλα σύνολα δεδομένων εκπαίδευσης.

Μπορεί να χειριστεί αποδοτικά πάρα πολύ μεγάλο αριθμό χαρακτηριστικών (ακόμα και χιλιάδες).

Δίνει μια εκτίμηση για το ποια χαρακτηριστικά είναι τα πιο σημαντικά στην κατηγοριοποίηση.

Δεν χρειάζεται την χρήση διαφορετικού συνόλου δεδομένων για τον έλεγχο ακρίβειας (δεν είναι δηλαδή απαραίτητο το cross-validation), καθώς η εκτίμηση του λάθους γενίκευσης γίνεται από τον ίδιο τον αλγόριθμο κατά την εκτέλεσή του.

Μπορεί να χειριστεί αποδοτικά ελλιπή δεδομένα.

Δεν παρουσιάζει φαινόμενα υπέρ εκπαίδευσης.

Ο ακριβής ορισμός του είναι: Ένα Random Forest είναι μας ταξινομητής που αποτελείται από μια συλλογή ταξινομητών μορφής δέντρων αποφάσεων. Κάθε δέντρο απόφασης αναπτύσσεται σε σχέση με ένα τυχαίο διάλυμα όπου $k=1\dots L$ είναι ανεξάρτητα και ισόνομα μεταξύ μας.

Κάθε δέντρο “ψηφίζει” για την πιο δημοφιλή κλάση εισόδου x .

Για να αποτραπεί ο αλγόριθμος να μάθει ξανά και ξανά το ίδιο δέντρο, η δειγματοληψία με αντικατάσταση από τα δεδομένα εκπαίδευσης και η τυχαία επιλογή των χαρακτηριστικών σε κάθε διαχωρισμό εξασφαλίζει ότι κάθε δέντρο στο τυχαίο δάσος είναι διαφορετικό. Ως αποτέλεσμα μας μας τυχαιότητας, η προκατάληψη του δάσους συνήθως ελαφρώς αυξάνεται (σε σχέση με τη μεροληψία μας μόνο δένδρου που βασίζεται σε στοιχεία που δεν έχουν δειγματοληψία) αλλά, λόγω του μέσου όρου, η διακύμανσή του μας μειώνεται, συνήθως περισσότερο από την αντιστάθμιση μας αύξησης μας μεροληψίας, αποδίδοντας έτσι ένα γενικότερο καλύτερο μοντέλο. Όταν κάνετε μια πρόβλεψη, η νέα παρατήρηση πιέζεται μας τα κάτω σε κάθε δέντρο απόφασης και εκχωρείται μια προβλεπόμενη τιμή.

Μόλις κάθε δέντρο του δάσους αναφέρει την προβλεπόμενη τιμή του, οι προβλέψεις κατατάσσονται στην τελική πρόβλεψη. Στην περίπτωση **δέντρων αποφάσεων**, η απλή πλειοψηφία (mode) καθορίζει την παραγωγή, ενώ στην περίπτωση μας **δένδρου παλινδρόμησης**, οι μέσες προβλέψεις όλων των επιμέρους δένδρων αποτελούν την τελική πρόβλεψη.

Ένα τυχαίο δάσος με: $N = 3$

Τυπικά,

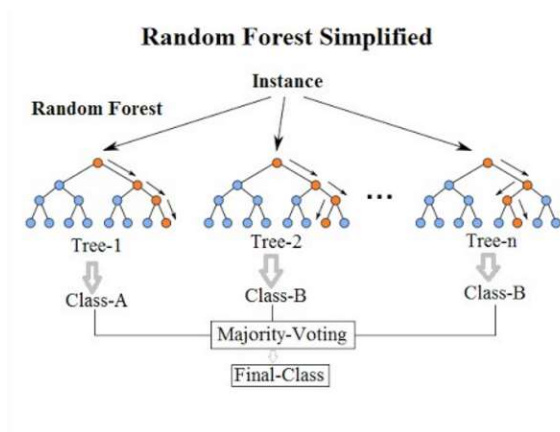
N

είναι πολύ υψηλότερο.

Γενικά, όσο μικρότερη είναι η συσχέτιση μεταξύ δέντρων, τόσο καλύτερη είναι η πρόβλεψη. Τα τυχαία δάση αποφάσεων διορθώνουν την τάση των δέντρων αποφάσεων να υπερκεράσουν το εκπαιδευτικό σύνολο μας. Αν κάποιο από τα δέντρα δημιουργεί σφάλματα ή τουλάχιστον θορυβώδη αποτελέσματα, άλλα δέντρα θα ξεπεράσουν τα δέντρα και θα ακυρώσουν τα λάθη.

Κάθε δέντρο κατασκευάζεται βάσει του ακόλουθου αλγόριθμου:

1. Έστω N ο αριθμός των περιπτώσεων εκπαίδευσης και M ο αριθμός των μεταβλητών στον ταξινομητή.
2. Ο αριθμός m των μεταβλητών εισόδου χρησιμοποιείται για να παρθεί απόφαση σε ένα κόμβο του δέντρου. $M \gg m$
3. Χρησιμοποιώντας μας υπόλοιπες υποθέσεις υπολογίζουμε το σφάλμα του δέντρου, προβλέποντας μας κλάσεις του.
4. Για κάθε κόμβο στο δέντρο επιλέγουμε τυχαία m μεταβλητές μας οποίες στηρίζουμε την απόφαση του κόμβου.
5. Υπολογίζουμε τον καλύτερο δυνατό διαχωρισμό βασισμένο μας m μεταβλητές του σετ εκπαίδευσης.



Εικόνα 43

4.5.9 Τεχνητά Νευρωνικά Δίκτυα (Artificial Neural Network-ANN)

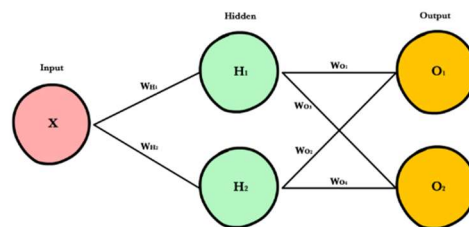
Ορισμός: Ένα Τεχνητό Νευρωνικό Δίκτυο είναι ένα υπολογιστικό μοντέλο βασισμένο στη δομή και μας συσχετίσεις των βιολογικών νευρωνικών δικτύων. Οι πληροφορίες που ρέουν μέσα από το δίκτυο επηρεάζουν τη δομή του, δηλαδή κατά μία έννοια το νευρωνικό δίκτυο “μαθαίνει” βάση των δεδομένων εισόδου- εξόδου.

Τα ANN είναι ένα πολύ χρήσιμο και ευέλικτο στατιστικό εργαλείο λόγω μας προσαρμοστικότητάς μας. Θεωρούνται ως μη γραμμικά στατιστικά μοντέλα ανάλυσης

δεδομένων όπου οι πολύπλοκες σχέσεις μεταξύ εισόδου και εξόδου μοντελοποιούνται και ανακαλύπτονται.

Τα σημαντικότερα πλεονεκτήματα των νευρωνικών δικτύων που πρέπει να τονίσουμε είναι η ικανότητά μας να «εκπαιδεύονται» παρακολουθώντας και αναλύοντας τα δεδομένα μας. Αυτά τα εργαλεία μας βοηθούν να υπολογίσουμε τη βέλτιστη μέθοδο λύσης καθώς ορίζουν υπολογιστικές συναρτήσεις και κατανομές βασισμένα μόνο σε ένα μέρος των δεδομένων μας παρά στο σύνολό μας. Τα ANN αποτελούνται από τρία αλληλένδετα επίπεδα.

Το πρώτο αποτελείται από την είσοδο των νευρώνων όπου και εισάγουμε τα δεδομένα μας (INPUT), τα οποία στέλνουν δεδομένα στο 2^ο επίπεδο (hidden layer) για ανάλυση και με τη σειρά μας στέλνουν τη μεταβλητή εξόδου στο 3^ο επίπεδο (output) που μας δίνει την πρόβλεψη (raw prediction).



Εικόνα 44

4.6 Προσέγγιση – Πρόβλεψη (Regression)

Στην περίπτωση της πρόβλεψης, έχουμε συνεχή δεδομένα. Θεωρούμε ότι οι έξοδοι των δεδομένων, t_i , είναι αποτελέσματα μιας συνάρτησης που εφαρμόζεται πάνω στις εισόδους μας, την αναλυτική όμως μορφή της συνάρτησης δεν την γνωρίζουμε. Χρησιμοποιώντας λοιπόν τα γνωστά δεδομένα μας, (x_i, t_i) , εκπαιδεύουμε έναν αλγόριθμο έτσι ώστε η συμπεριφορά του να προσεγγίζει την άγνωστη σε εμάς συνάρτηση. Όταν του εισάγουμε τις γνωστές μας τιμές να μας δίνει τις σωστές εξόδους και όταν εισάγουμε νέες τιμές με άγνωστες τιμές εξόδου, αυτός να μπορεί να προβλέψει την τιμή που θα έδινε σαν έξοδο η άγνωστη συνάρτηση.

Ουσιαστικά είναι γενίκευση των αντίστοιχων μεθόδων πρόβλεψης της αριθμητικής ανάλυσης, για παράδειγμα μπορεί να εφαρμοστεί και σε μορφές δεδομένων στις οποίες δεν μπορούμε να εφαρμόσουμε τις κλασικές μεθόδους.

Η μορφή της συνάρτησης που προσπαθούμε να προσεγγίσουμε παραμένει άγνωστη και η ακρίβεια της πρόβλεψης εξαρτάται από πολλούς παράγοντες.

Κεφάλαιο 5: Υλοποίηση

Για να χρησιμοποιήσουμε δεδομένα κειμένου σε έναν αλγόριθμο ml πρέπει να τα μετατρέψουμε σε αριθμητικά διανύσματα χαρακτηριστικών με σταθερό μέγεθος, επειδή οι περισσότεροι αλγόριθμοι δεν μπορούν να χρησιμοποιήσουν ακατέργαστα έγγραφα κειμένου με διαφορετικό μήκος. Για να τα μετατρέψουμε εύκολα πρέπει να χρησιμοποιήσουμε μια

βιβλιοθήκη μηχανικής μάθησης (ml). Για να την υλοποίησή της λύση μας χρησιμοποιήθηκε η βιβλιοθήκη Scikit-learn καθώς και η γλώσσα προγραμματισμού Python

5.1 Τι είναι η γλώσσα προγραμματισμού Python



Εικόνα 45

Η Python είναι μια διερμηνευμένη, αντικειμενοστραφής, υψηλού επιπέδου γλώσσα προγραμματισμού με δυναμική σημασιολογία. Οι ενσωματωμένες δομές δεδομένων υψηλού επιπέδου σε συνδυασμό με τη δυναμική τυποποίηση και τη δυναμική δέσμευση την καθιστούν ιδανική για την ταχεία ανάπτυξη εφαρμογών και ως γλώσσα σεναρίων ή συνδέσμων για τη σύνδεση υφιστάμενων στοιχείων μεταξύ τους. Η εύκολη στην εκμάθηση σύνταξη της Python δίνει έμφαση στην αναγνωσιμότητα, μειώνοντας έτσι το κόστος συντήρησης του προγράμματος. Η Python υποστηρίζει ενότητες και πακέτα, τα οποία προωθούν την αρθρωτότητα των προγραμμάτων και την επαναχρησιμοποίηση του κώδικα.

Ο διερμηνέας της Python και η εκτεταμένη πρότυπη βιβλιοθήκη είναι ελεύθερα διαθέσιμες σε όλες τις κύριες πλατφόρμες σε πηγαία ή δυαδική μορφή και μπορούν να διανεμηθούν ελεύθερα.

5.2 Τι είναι η βιβλιοθήκη Scikit-learn



Εικόνα 46

Η Scikit-learn (Sklearn) είναι η πιο χρήσιμη και ισχυρή βιβλιοθήκη μηχανικής μάθησης στην Python. Παρέχει μια σειρά από αποδοτικά εργαλεία μηχανικής μάθησης και στατιστικής μοντελοποίησης, όπως ταξινόμηση, παλινδρόμηση, ομαδοποίηση και μείωση διαστάσεων,

μέσω μιας συνεπούς διεπαφής στην Python. Αυτή η βιβλιοθήκη είναι γραμμένη κυρίως σε Python και βασίζεται στις NumPy, SciPy και Matplotlib.[16]

5.3 Πως λειτουργεί ο κώδικας

Για να μπορέσουμε να χρησιμοποιήσουμε την πρόταση στη μηχανική μάθηση η πρόταση πρέπει να περαστεί μέσα από το μετατροπέα ομιλίας σε κείμενο της google ώστε να μπορέσουμε να τη χρησιμοποιήσουμε στα εργαλεία για την επεξεργασία κειμένου που μας παρέχει η βιβλιοθήκη της Scikit-learn. Αφού αποσπάσουμε το κείμενο από τον ήχο πρέπει να το περάσουμε μέσα από τα εργαλεία που μας παρέχει η Scikit-learn ώστε να μπορέσουν να χρησιμοποιηθούν στον αλγόριθμο μηχανικής μάθησης.

Σε αυτή την πτυχιακή χρησιμοποιούμε το μοντέλο bag of words. Εν συντομία, χωρίζουμε κάθε αρχείο κειμένου σε λέξεις και μετράμε τον αριθμό των φορών που κάθε λέξη βρίσκεται σε κάθε έγγραφο και αναθέτουμε σε κάθε λέξη ένα ακέραιο αναγνωριστικό.

Κάθε μοναδική λέξη στο λεξικό μας θα αντιστοιχεί σε ένα χαρακτηριστικό.

5.4 Τι είναι το μοντέλο bag of words

Bag of words (BoW)

Very good drama although it appeared to have a few blank areas leaving the viewers to fill in the action for themselves. I can imagine life being this way for someone who can neither read nor write. This film simply smacked of the real world: the wife who is suddenly the sole supporter, the live-in relatives and their quarrels, the troubled child who gets knocked up and then, typically, drops out of school, a jackass husband who takes the nest egg and buys beer with it. 2 thumbs up... very very very good movie.



('the', 8),
(',', 5),
('very', 4),
('.', 4),
('who', 4),
('and', 3),
('good', 2),
('it', 2),
('to', 2),
('a', 2),
('for', 2),
('can', 2),
('this', 2),
('of', 2),
('drama', 1),
('although', 1),
('appeared', 1),
('have', 1),
('few', 1),
('blank', 1)
.....

Εικόνα 47

Το μοντέλο bag-of-words είναι μια μέθοδος εξαγωγής χαρακτηριστικών από κείμενο για σκοπούς μοντελοποίησης, όπως η χρήση αλγορίθμων μηχανικής μάθησης.

Η μέθοδος είναι τόσο απλή και ευέλικτη που τα χαρακτηριστικά μπορούν να εξαχθούν από έγγραφα με άπειρο αριθμό τρόπων.

Μια σακούλα λέξεων είναι μια αναπαράσταση κειμένου που περιγράφει τις λέξεις που εμφανίζονται σε ένα έγγραφο. Πρόκειται για δύο πράγματα:

- Γνωστό λεξιλόγιο.
- Μέτρηση της παρουσίας γνωστών λέξεων.

Ονομάζεται "σάκος λέξεων" επειδή απορρίπτονται όλες οι πληροφορίες σχετικά με τη σειρά ή τη δομή των λέξεων στο έγγραφο. Το μοντέλο δίνει προσοχή μόνο στο αν μια γνωστή λέξη εμφανίζεται στο έγγραφο και όχι στο πού στο έγγραφο.

Οι σακούλες λέξεων μπορούν να είναι όσο απλές ή σύνθετες θέλουμε. Η πολυπλοκότητα προκύπτει από την απόφαση για το πώς θα σχεδιαστεί το λεξιλόγιο των γνωστών λέξεων (ή tokens) και πώς θα αξιολογηθεί η ύπαρξη γνωστών λέξεων.

5.5 Ανάλυση κώδικα του κυρίως προγράμματος

Στον παρακάτω πίνακα βλέπουμε τις βιβλιοθήκες που θα πρέπει εισάγουμε

```
import speech_recognition as sr
from kalliope.core import Utils
from kalliope.stt.Utils import SpeechRecognition
from sklearn import datasets
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.feature_extraction.text import TfidfTransformer
from sklearn.naive_bayes import MultinomialNB
from sklearn.feature_extraction.text import TfidfVectorizer
import numpy as np
import os
import re
```

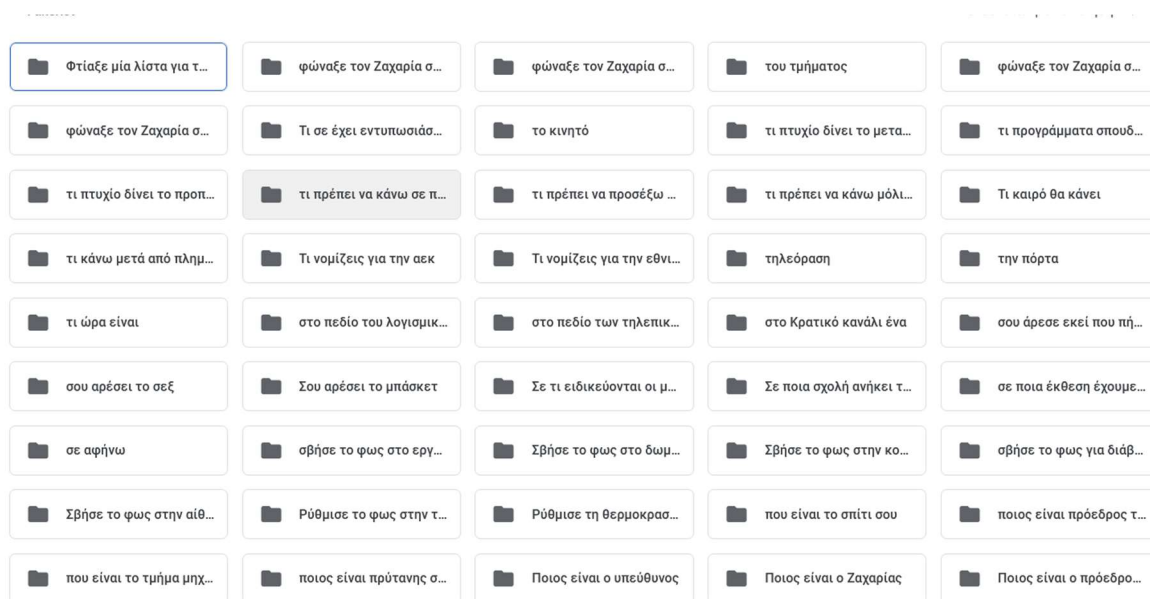
Πίνακας 3

Στην συνέχεια, μεταφέρουμε το κείμενο από το speech to text αλγόριθμο της google στο δικό μας κομμάτι .

```
def google_callback(self, recognizer, audio):
    ans= self.ml(captured_audio)
    Utils.print_success("ml thinks you said %s" %ans)
```

Πίνακας 4

Επιπλέον, εντάσσουμε όλα τα δεδομένα τα οποία θα χρησιμοποιηθούν για τον αλγόριθμο μηχανικής μάθησης το οποία είναι τα εξής.



Εικόνα 48

```
def ml(self, dataml):
    root="/home/a/kati_kal/kati/resources/stt/googlemod/chatbot/chatbot-train"
    self.cases=sklearn.datasets.load_files(root, description=None,
categories=None, load_content=True,shuffle=True, encoding='utf-8',
decode_error='strict', random_state=0)
```

Πίνακας 5

Ο αλγόριθμος μαθαίνει το λεξιλόγιο και επιστρέφει έναν πίνακα Document-term. (tokenization).

```
count_vect = CountVectorizer()
X_train_counts = count_vect.fit_transform(self.cases.data)
X_train_counts.shape
```

Πίνακας 6

Προκειμένου να επιλύσουμε πιθανές διαφορές από έγγραφα μεγαλύτερης διάρκειας, διαιρούμε τον αριθμό των εμφανίσεων κάθε λέξης σε ένα έγγραφο με τον συνολικό αριθμό των λέξεων (Term Frequency).

Επίσης, υποβαθμίζουμε και το βάρος των λέξεων που εμφανίζονται πολλές φορές και δεν περιέχουν καμία πρόσθετη πληροφορία σε ένα έγγραφο (Term Frequency επί Inverse Document Frequency).

```

tfidf_transformer = TfidfTransformer()
X_train_tfidf = tfidf_transformer.fit_transform(X_train_counts)
X_train_tfidf.shape

```

Πίνακας 7

Εκπαιδεύουμε τον κατηγοριοποιητή μας στην περίπτωση μας MultinomialNB.

```

clf = MultinomialNB(alpha=0.1).fit(X_train_tfidf, self.cases.target)

```

Πίνακας 8

Χρησιμοποιούμε και πάλι τις μεθόδους που αναφέραμε προηγουμένως για τη φράση μας.

```

self.docs_new = [dataml]
X_new_counts = count_vect.transform(self.docs_new)
X_new_tfidf = tfidf_transformer.transform(X_new_counts)
tf=X_new_tfidf
self.predicted = clf.predict(X_new_tfidf)

```

Πίνακας 9

Αποθηκεύει τις μεταβλητές μας όταν εντοπίζει μια από αυτές τις λέξεις.

```

var=[0,0]
if "να πάρω " in dataml or "ύμισε μου" in dataml or "ύμισέ μου" in dataml :
    var = self.savevar(dataml)
return self.apotel(clf,tf,categlen,var)

```

Πίνακας 10

Ελέγχει αν η φράση μας είναι άγνωστη ή όχι και επιστρέφει άγνωστο αν δεν υπάρχει.

```

def __str__(self):
    return self.answ.lower()
def apotel(self,clf,tf,categlen,var):
    a=clf.predict_proba(tf)
    ma=float(a.max())
    freq=np.count_nonzero(a == ma)

```

```
if freq > 1 :
    self.answ="unknown"
    return self.answ
```

Πίνακας 11

Αποθηκεύει την προβλεπόμενη λύση μας στη μεταβλητή answ.

```
for doc, category in zip(self.docs_new, self.predicted):
    self.answ=self.cases.target_names[category]
```

Πίνακας 12

Αν τα δεδομένα μας είχαν μεταβλητές, αντικαθιστά τις μεταβλητές `{{ remember }}` και `{{ time }}` με τις σωστές μεταβλητές.

```
if "{{ remember }}" in self.answ :
    self.answ=self.answ.replace("{{ remember }}",var[0])
    self.answ=self.answ.replace("{{ time }}",var[1])
return self.answ
```

Πίνακας 13

Βρίσκει το μοτίβο μας και αποθηκεύει την πρώτη μας μεταβλητή. Αν δεν βρει το σωστό μοτίβο προσθέτει ως μεταβλητή το "κατι".

```
def savevar(self,dataml):
    var=['','']
    pattern = r"^(?P<thing>.*?) το (?P<rem>.*?) .*?$"
    match = re.match(pattern, dataml)
    try:
        var[0]=match.group("rem")
    except:
        var[0]='κατι'
```

Πίνακας 14

Βρίσκει τον αριθμό στη φράση μας και τον αποθηκεύει. Αν δεν βρει κανέναν αριθμό, αποθηκεύει 60.

```
var[1]=re.sub("[^0-9]", "", dataml)
if var[1] == "":
    var[1]='60'
```

```
return var
```

Πίνακας 15

5.6 Πως μπορούμε να προσθέσουμε καινούργιες προτάσεις

Για να προσθέσουμε καινούργιες προτάσεις με στόχο να τις χρησιμοποιήσουμε θα πρέπει να τις προσθέσουμε στο φάκελο με τις προτάσεις.

Αυτό μπορούμε να το κάνουμε είτε χειροκίνητα είτε μέσω ενός προγράμματος. Για να μην γίνεται χειροκίνητα δημιουργήσαμε το script που θα γίνει ανάλυση στην συνέχεια της διπλωματικής εργασίας.

Αρχικά για να εκτελέσουμε το πρόγραμμα τρέχουμε την έξης εντολή:

- `python3 nameOfScript`, στο φάκελο στον οποίο έχει δημιουργηθεί το αρχικό μας πρόγραμμα με όνομα `input.py`.

```
79 : τι ώρα είναι
80 : τι κάνω μετά από πλημμύρα
81 : Τι καιρό θα κάνει
82 : Τι νομίζεις για την αεκ
83 : Τι νομίζεις για την εθνική με την κροατία
84 : τι πρέπει να κάνω μόλις δω φωτιά
85 : τι πρέπει να κάνω σε πλημμύρα
86 : τι πρέπει να προσέξω όταν έχει πλημμύρα
87 : τι προγράμματα σπουδών έχει το τμήμα μας
88 : τι πτυχίο δίνει το μεταπτυχιακό πρόγραμμα
89 : τι πτυχίο δίνει το προπτυχιακό πρόγραμμα
90 : Τι σε έχει εντυπωσιάσει στο Ελληνικό ποδόσφαιρο
91 : το κινητό
92 : του τμήματος
93 : φώναξε τον Ζαχαρία στην κουζίνα
94 : φώναξε τον Ζαχαρία στο κρεβάτι
95 : φώναξε τον Ζαχαρία στο τραπέζι
96 : φώναξε τον Ζαχαρία στον καναπέ
97 : Φτιάξε μία λίστα για τα ψώνια
(1) do you want to add a new phrase on a current command (folder)
(2) do you want to add a new command (folder)
(3)do you want to rename a command (folder)
```

Εικόνα 49

Και ακολουθούμε τις εντολές.

Αν θέλουμε να προσθέσουμε μια καινούργια φράση σε μια εντολή που είδη έχουμε προσθέσει πατάμε το 1 και πληκτρολογούμε τον αριθμό φακέλου που έχει στην αρχή.

```
1
which command (folder) ?
```

Εικόνα 50

Στην συνέχεια, μας ζητάει να πληκτρολογήσουμε την καινούργια πρόταση στο φάκελο που επιλέξαμε.

```
1
the name of the command (folder) is : άναψε το κλιματιστικό στην κουζίνα
type another phrase for the command(folder)
```

Εικόνα 51

Για να προσθέσουμε μια καινούργια εντολή πληκτρολογούμε το 2 και πληκτρολογούμε την καινούργια εντολή.

```
(3)do you want to rename a command (folder)
2
Type a new command:
```

Εικόνα 52

Ωστόσο, αν θέλουμε να αλλάξουμε κάποιο όνομα από μια υπάρχουσα ερώτηση πατα το 3 και πληκτρολογούμε τον αριθμό που μας βγάζει.

```
(3)do you want to rename a command (folder)
3
which command (folder) do you want to rename :(number)
2
```

Εικόνα 53

Μετά μας ζητάει να πληκτρολογήσουμε τη νέα ονομασία που θέλουμε να έχει η εντολή

```
the name of the command (folder) is : άναψε το κλιματιστικό στο εργαστήριο
type the new name
```

Εικόνα 54

5.7 Ανάλυση κώδικα του προγράμματος ένταξης προτάσεων

Εδώ βλέπουμε τις βιβλιοθήκες που θα πρέπει εισάγουμε και τα μονοπάτια των αρχείων

```
#!/usr/bin/env python3
from pathlib import Path
import random
import os
class inp:
    def main(self):
```

```

        self.directory="C:\Users\vasil\Downloads\kati\kati\resources\stt\googlemod\
chatbot\directory"

        self.root="C:\Users\vasil\Downloads\kati\kati\resources\stt\googlemod\chatb
ot\chatbot-train"

```

Πίνακας 16

Βρίσκει και αποθηκεύει τα ονόματα των φακέλων. Αν η εντολή μας έχει ήδη προστεθεί μας ρωτάει αν θέλουμε να προσθέσουμε άλλη μια φράση.

```

while True:

    samedir=""

    dirlist = [ item for item in os.listdir(self.root) if
os.path.isdir(os.path.join(self.root, item)) ]

    exit = input("Do you want to add a new command? yes/no\n")

    if exit != "yes" :

        break

    txt = input("Type a new command:\n")

    if txt in dirlist:

        secondprase = input("This command already exists do you want to add
another phrase in it? yes/no\n")

        if secondprase == "yes":

            samedir=txt

            txt = input("add the second phrase for the command\n")

            self.adding(samedir,txt)

        else:

            self.adding(samedir,txt)

```

Πίνακας 17

Αποθηκεύει την τρέχουσα διαδρομή μας αν θέλουμε να προσθέσουμε μια άλλη φράση. Ελέγχει αν ο φάκελος υπάρχει, και αν δεν υπάρχει τον δημιουργεί καθώς επίσης φτιάχνει ένα νέο αρχείο με τη νέα μας φράση και το αποθηκεύει με έναν τυχαίο αριθμό.

```

def adding(self,samedir,txt):

    if samedir != "":

        x=self.root+samedir+"/"

    else:

```

```

        x=self.root+txt+"/"

print("This is what you typed ", txt)

Path(x).mkdir(parents=True, exist_ok=True)

n = random.randint(0,30000)

r=str(n) + ".txt"

file = open(x + r, "w")

file.write(txt)

file = open(x + r, "r")

```

Πίνακας 18

Βρίσκει την τελευταία μας γραμμή στο αρχείο καταλόγου και γράφει τη νέα μας φράση και τον τυχαίο αριθμό σε αυτήν.

```

with open(self.directory, "r") as file:
    for last_line in file:
        pass
    print("last line was : "+last_line)
    file = open(self.directory, "a")
    if samedir != "":
        file.write('\n'+str(n)+"\tmain commands is :"+samedir+" and the
added new phrase is : "+txt)
    else:
        file.write('\n'+str(n)+"\t"+txt)
    file.close()

if __name__ == "__main__":
    objName = inp()
    objName.main()

```

Πίνακας 19

Κεφάλαιο 6: Συμπεράσματα και μελλοντικές επεκτάσεις

Αυτή η πτυχιακή είχε σαν σκοπό να γίνει επέκταση των λειτουργιών της Αλεξάνδρας ώστε να μπορέσει ο χώρος της έξυπνης οικίας να βοηθήσει καλύτερα τους ηλικιωμένους ανθρώπους όπου δεν είναι καθόλου απίθανο να ξεχναγαν καμιά φορά τις ακριβές εντολές που είχαν δώσει στην Αλεξάνδρα. Κάποιες από τις μελλοντικές επεκτάσεις που θα μπορούσαν να υπάρξουν είναι η συλλογή δεδομένων από το internet ώστε να μπορέσει να επεκταθεί το ήδη υπάρχων dataset ώστε να γίνει καλύτερη η αναγνώριση των ήδη υπάρχων προτάσεων.

Αναφορές

- [1] «<https://www.apple.com/siri/>,» [Ηλεκτρονικό].
- [2] «<https://assistant.google.com/>,» [Ηλεκτρονικό].
- [3] «<https://support.microsoft.com/el-gr/topic/what-is-cortana-953e648d-5668-e017-1341-7f26f7d0f825>,» [Ηλεκτρονικό].
- [4] «Oracle,» [Ηλεκτρονικό]. Available: <https://www.oracle.com/chatbots/what-is-a-chatbot/>.
- [5] «<https://en.wikipedia.org/wiki/ELIZA>,» [Ηλεκτρονικό].
- [6] «<https://en.wikipedia.org/wiki/PARRY>,» [Ηλεκτρονικό].
- [7] «<https://en.wikipedia.org/wiki/Jabberwacky>,» [Ηλεκτρονικό].
- [8] «https://en.wikipedia.org/wiki/Dr._Sbaitso,» [Ηλεκτρονικό].
- [9] «https://en.wikipedia.org/wiki/Artificial_Linguistic_Internet_Computer_Entity,» [Ηλεκτρονικό].
- [10] «<https://en.wikipedia.org/wiki/SmarterChild>,» [Ηλεκτρονικό].
- [11] «<https://www.amazon.com/b?ie=UTF8&node=21576558011>,» [Ηλεκτρονικό].
- [12] L. M. Eleni Adamopoulou, "Chatbots: History, technology, and applications," 15 December 2020. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S2666827020300062#b172>.
- [13] "Kalliope doc," [Online]. Available: <https://kalliope-project.github.io/kalliope/>.
- [14] <https://www.loxone.com/enen/commercial/ambient-assisted-living-aal/>
- [15] <https://www.python.org/doc/essays/blurb/>
- [16] https://www.tutorialspoint.com/scikit_learn/index.htm