



**ΠΑΝΕΠΙΣΤΗΜΙΟ ΔΥΤΙΚΗΣ ΑΤΤΙΚΗΣ**  
**ΣΧΟΛΗ ΜΗΧΑΝΙΚΩΝ**  
**ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ**  
**ΥΠΟΛΟΓΙΣΤΩΝ**

**ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ**

**Ιατρικός φάκελος με δυνατότητα πρόβλεψης διαβήτη**

**Βασίλειος Πεσκελίδης**

**Εισηγητής: Δρ Νικόλαος Ζάχαρης, Καθηγητής**

**ΑΘΗΝΑ**

# Ιατρικός φάκελος με δυνατότητα πρόβλεψης διαβήτη

2018

**Ιατρικός φάκελος με δυνατότητα πρόβλεψης διαβήτη**

**ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ**

**Ιατρικός φάκελος με δυνατότητα πρόβλεψης διαβήτη**

**Βασίλειος πεσκελίδης  
Α.Μ. 43018**

**Εισηγητής:**

**Δρ Νικόλαος Ζάχαρης, Καθηγητής**

**Εξεταστική Επιτροπή:**

**Γιαννακόπουλος, Παναγιώτης  
Πρεζεράκος, Γεώργιος**

**Ημερομηνία εξέτασης**

## Ιατρικός φάκελος με δυνατότητα πρόβλεψης διαβήτη

## **ΔΗΛΩΣΗ ΣΥΓΓΡΑΦΕΑ ΠΤΥΧΙΑΚΗΣ ΕΡΓΑΣΙΑΣ**

Ο/Η κάτωθι υπογεγραμμένος/η ΠΕΣΚΕΛΙΔΗΣ ΒΑΣΙΛΕΙΟΣ,  
Του ΑΛΕΞΑΝΔΡΟΥ, με αριθμό μητρώου 43018 φοιτητής/τρια του Τμήματος  
Μηχανικών Η/Υ Συστημάτων Τ.Ε. του Α.Ε.Ι. Πειραιά Τ.Τ. πριν αναλάβω την  
εκπόνηση της Πτυχιακής Εργασίας μου, δηλώνω ότι ενημερώθηκα για τα  
παρακάτω:

«Η Πτυχιακή Εργασία (Π.Ε.) αποτελεί προϊόν πνευματικής ιδιοκτησίας τόσο του  
συγγραφέα, όσο και του Ιδρύματος και θα πρέπει να έχει μοναδικό χαρακτήρα και  
πρωτότυπο περιεχόμενο.

Απαγορεύεται αυστηρά οποιοδήποτε κομμάτι κειμένου της να εμφανίζεται  
αυτούσιο ή μεταφρασμένο από κάποια άλλη δημοσιευμένη πηγή. Κάθε τέτοια  
πράξη αποτελεί προϊόν λογοκλοπής και εγείρει θέμα Ηθικής Τάξης για τα  
πνευματικά δικαιώματα του άλλου συγγραφέα. Αποκλειστικός υπεύθυνος είναι ο  
συγγραφέας της Π.Ε., ο οποίος φέρει και την ευθύνη των συνεπειών, ποινικών και  
άλλων, αυτής της πράξης.

Πέραν των όποιων ποινικών ευθυνών του συγγραφέα σε περίπτωση που το  
Ίδρυμα του έχει απονεμίσει Πτυχίο, αυτό ανακαλείται με απόφαση της Συνέλευσης  
του Τμήματος. Η Συνέλευση του Τμήματος με νέα απόφαση της, μετά από  
αίτηση του ενδιαφερόμενου, του αναθέτει εκ νέου την εκπόνηση της Π.Ε. με άλλο  
θέμα και διαφορετικό επιβλέποντα καθηγητή. Η εκπόνηση της εν λόγω Π.Ε.  
πρέπει να ολοκληρωθεί εντός τουλάχιστον ενός ημερολογιακού 6μήνου από την  
ημερομηνία ανάθεσης της. Κατά τα λοιπά εφαρμόζονται τα προβλεπόμενα στο  
άρθρο 18, παρ. 5 του ισχύοντος Εσωτερικού Κανονισμού.»

## Ιατρικός φάκελος με δυνατότητα πρόβλεψης διαβήτη

## **ΕΥΧΑΡΙΣΤΙΕΣ**

Η παρούσα πτυχιακή εργασία ολοκληρώθηκε μετά από επίμονες προσπάθειες, σε ένα ενδιαφέρον γνωστικό αντικείμενο, όπως αυτό του διαδικτυακού προγραμματισμού και τις μηχανικής μάθησης. Την προσπάθειά μου αυτή υποστήριξε ο επιβλέπων καθηγητής μου, τον οποίο θα ήθελα να ευχαριστήσω.

Ακόμα Θα ήθελα να ευχαριστήσω την οικογένεια μου και τους φίλους μου για τη βοήθεια που μου προσέφεραν καθώς χωρίς αυτούς δε θα μπορούσα να καταφέρω τα επιθυμητά αποτελέσματα.

## Ιατρικός φάκελος με δυνατότητα πρόβλεψης διαβήτη



## ΠΕΡΙΛΗΨΗ

Η παρούσα πτυχιακή εργασία ασχολείται με τη δημιουργία διαδικτυακής εφαρμογής για καταγραφή ιστορικού ασθενή και χορήγηση φαρμάκων από ιατρό με δυνατότητα πρόβλεψης διαβήτη καθώς και κλήση ραντεβού από ασθενή.

Η σελίδα για χορήγηση και καταγραφή είναι αποκλειστικά για εξουσιοδοτημένους ιατρούς και προσωπικό. Ενώ τα ραντεβού είναι για όσους είναι εγγραμμένα στο σύστημα.

Τέλος το κομμάτι της πρόβλεψης αποτελεί κλάδο της τεχνητής νοημοσύνης όπου και εφαρμόζονται αλγόριθμοι μηχανικής μάθησης.

## ABSTRACT

The present thesis concerns the development of web application for storing patient data and drug prescription from a doctor with abilities of predicting diabetes. Moreover patients can arrange an appointment via the web site.

The page for storing and prescription is for authorized doctors and staff. While the appointment arranging is for the public as long as he or she is registered in the system.

Finally the predicting part is a field of artificial intelligence where machine learning algorithms are applied.

# Ιατρικός φάκελος με δυνατότητα πρόβλεψης διαβήτη

ΕΠΙΣΤΗΜΟΝΙΚΗ ΠΕΡΙΟΧΗ: Σχεδιασμός ιστοσελίδας, Τεχνητή νοημοσύνη  
ΛΕΞΕΙΣ ΚΛΕΙΔΙΑ: διαδικτυακή εφαρμογή, ιστοσελίδα, μηχανική μάθηση

## ΠΕΡΙΕΧΟΜΕΝΑ

<b>ΚΕΦΑΛΑΙΟ 1</b> .....	<b>19</b>
<b>Διαδίκτυο και τεχνολογίες</b> .....	<b>19</b>
ΕΙΣΑΓΩΓΗ.....	19
1.1 Ιστορική αναδρομή.....	19
1.2 Μοντέλο TCP/IP .....	20
1.3 URL και HTTP.....	20
1.4 Πρωτόκολλο DNS .....	21
1.5 Πρωτόκολλο TCP και UDP .....	21
1.6 Πρωτόκολλο IP .....	21
1.7 εξυπηρετητής (server).....	22
<b>ΚΕΦΑΛΑΙΟ 2</b> .....	<b>23</b>
<b>Εργαλεία και γλώσσες ανάπτυξης</b> .....	<b>23</b>
Εισαγωγή .....	23
2.1 HTML .....	23
2.2 PHP.....	24
2.3 JavaScript .....	25
2.4 CSS.....	25
2.5 MySQL .....	26
2.6 Python.....	26
2.7 UML .....	27
<b>ΚΕΦΑΛΑΙΟ 3</b> .....	<b>28</b>
<b>Ιατρικός φάκελος</b> .....	<b>28</b>
Εισαγωγή .....	28
3.1 Βαση δεδομένων.....	30
3.2 Ιστοσελίδα Ιατρού .....	33

## Ιατρικός φάκελος με δυνατότητα πρόβλεψης διαβήτη

3.2.1 Φόρμα εισόδου .....	34
3.2.2 Ιατρική σελίδα .....	37
3.3 Σελίδα ασθενη .....	44
3.3.1 Φορμα εισοδου .....	44
3.3.2 Φόρμα ασθενή .....	44
3.4 Φόρμα γραμματείας .....	49
3.4.1 Φόρμα εισόδου .....	49
3.4.2 Φόρμα γραμματείας .....	50
<b>ΚΕΦΑΛΑΙΟ 4.....</b>	<b>52</b>
<b>Πρόβλεψη διαβήτη.....</b>	<b>52</b>
Εισαγωγή .....	52
4.1 Ιστορική αναδρομή.....	52
4.2 Μηχανική μάθηση .....	54
4.3 Πρόβλεψη διαβήτη .....	58
4.3.1 Gaussian Naive Bayes.....	59
4.3.2 SVM (SupportVectorMachine) .....	62
4.3.3 Random forest.....	65
4.3.4 Βιβλιοθήκες εγκατάστασης.....	69
<b>ΚΕΦΑΛΑΙΟ 5.....</b>	<b>70</b>
<b>Συμπεράσματα .....</b>	<b>70</b>
<b>ΒΙΒΛΙΟΓΡΑΦΙΑ.....</b>	<b>71</b>

## ΚΑΤΑΛΟΓΟΣ ΕΙΚΟΝΩΝ

Εικόνα 2.1 Παράδειγμα κωδικά HTML .....	24
Εικόνα 2.2 Παράδειγμα κωδικά HTML με PHP .....	24
Εικόνα 2.3 Παράδειγμα κωδικά javascript.....	25
Εικόνα 3.1 UML διάγραμμα περιπτώσεων χρήσης.....	28
Εικόνα 3.2 Σχεσιακό μοντέλο της βάσης.....	30
Εικόνα 3.3 Σύνταξη τις INNER JOIN .....	31
Εικόνα.3.4 SQL κώδικας για το ιστορικό του ασθενή .....	32
Εικόνα.3.5 SQL κώδικας για εμφάνιση προγραμματισμένων ραντεβού.....	32
Εικόνα.3.6 Σύνδεση με τη βάση .....	33
Εικόνα.3.7 Αποτέλεσμα ερωτήματος.....	33
Εικόνα.3.8 Προσπέλαση εγγραφών .....	33
Εικόνα 3.6 Φόρμα εισόδου.....	34
Εικόνα 3.7 Κώδικας φόρμας εισόδου .....	34
Εικόνα 3.8 Κώδικας CSS .....	35
Εικόνα 3.9 Κώδικας επιλογής γλώσσας και εξωτερικού αρχείου .....	35
Εικόνα 3.10 Κώδικας ελέγχου δεδομένων .....	36
Εικόνα 3.11 Κώδικας εμφάνισης μενού κουμπιού.....	37
Εικόνα 3.12 Δεδομένα Ιατρού .....	38
Εικόνα 3.13 Κώδικας μορφοποίησης πλαισίου .....	39
Εικόνα 3.14 Περιεχόμενο αναζήτησης.....	39
Εικόνα 3.15 Κώδικας περιεχομένου αναζήτησης .....	39
Εικόνα 3.16 Πίνακας ιστορικού ασθενή.....	40
Εικόνα 3.17 Κώδικας ιστορικού ασθενή .....	40
Εικόνα 3.18 Ερώτημα για την αποθήκευση δεδομένων τις εξέτασης.....	41

## Ιατρικός φάκελος με δυνατότητα πρόβλεψης διαβήτη

Εικόνα 3.19 Συνάρτηση <code>sell_exec()</code> .....	42
Εικόνα 3.20 Python κώδικας μετατροπείς δεδομένων .....	43
Εικόνα 3.21 Συνθήκη για την εκτέλεση αλγορίθμου .....	43
Εικόνα 3.22 Κώδικας για την εμφάνιση της πρόβλεψης.....	43
Εικόνα 3.23 Φόρμα εισόδου ασθενή .....	44
Εικόνα 3.24 Κώδικας για τον έλεγχο δεδομένων .....	44
Εικόνα 3.25 Φόρμα ασθενή .....	45
Εικόνα 3.26 Φόρμα ραντεβού .....	45
Εικόνα 3.27 Κώδικας για τις διαθέσιμες ώρες .....	46
Εικόνα 3.28 UML διάγραμμα ακολουθίας ραντεβού .....	47
Εικόνα 3.29 Κώδικας για τα τρέχοντα ραντεβού .....	48
Εικόνα 3.30 Φόρμα εισόδου γραμματείας .....	49
Εικόνα 3.31 Κώδικας ελέγχου δεδομένων .....	49
Εικόνα 3.32 Σελίδα γραμματείας .....	50
Εικόνα 3.33 Μενού διαγραφείς δεδομένων .....	50
Εικόνα 3.34 Κώδικας διαγραφείς δεδομένων .....	50
Εικόνα 3.35 Μενού αναζήτησης πληροφοριών ασθενή .....	51
Εικόνα 3.36 Μενού εγγραφής ασθενή .....	51
Εικόνα 4.1 Τεχνητό νευρικό δίκτυο .....	56
Εικόνα 4.2 Άθροισμα εισόδων .....	56
Εικόνα 4.3 Συνάρτηση sigmoid .....	57
Εικόνα 4.4 Είσοδοι με βάρη σε κόμβους .....	57
Εικόνα 4.5 Sigmoid με python .....	57
Εικόνα 4.6 Φόρτωμα δεδομένων από εξωτερικό αρχείο .....	58
Εικόνα 4.7 Όπου $x$ εξαρτημένες μεταβλητές και $y$ ανεξάρτητες.....	58
Εικόνα 4.8 Δεδομένα προς εκπαίδευση και τεστάρισμα .....	58
Εικόνα 4.9 Μοντέλο του αλγορίθμου Gaussian Naive Bayes .....	59
Εικόνα 4.8 Συνάρτηση Gauss .....	59

## Ιατρικός φάκελος με δυνατότητα πρόβλεψης διαβήτη

Εικόνα 4.9 Εξίσωση Gauss .....	60
Εικόνα 4.10 διάγραμμα Gauss .....	60
Εικόνα 4.11 Πιθανότητες άπω την εξίσωση Gauss .....	61
Εικόνα 4.12 Αποτέλεσμα του Gaussian Naive Bayes .....	62
Εικόνα 4.13 Βέλτιστο διάνυσμα .....	63
Εικόνα 4.14 Margin .....	63
Εικόνα 4.15 LagrangeMultiplier .....	63
Εικόνα 4.16 διαχωρισμός δεδομένων .....	63
Εικόνα 4.17 Μοντέλο του αλγορίθμου SVM .....	64
Εικόνα 4.18 RBF kernel .....	64
Εικόνα 4.19 Polynomial kernel .....	64
Εικόνα 4.19 Δένδρο αποφάσεων .....	64
Εικόνα 4.20 Σύνολο δένδρων αποφάσεων .....	66
Εικόνα 4.21 Μέθοδος Gini .....	67
Εικόνα 4.22 Μοντέλο του αλγορίθμου Random Forest .....	69

## ΚΑΤΑΛΟΓΟΣ ΠΙΝΑΚΩΝ

Πίνακας 1.1 Κλάσεις IP διευθύνσεων .....	64
Πίνακας 1.2 Παράδειγμα IP διεύθυνσης 147.10.13.28 .....	64
Πίνακας 4.1 Χωρισμός δεδομένων.....	66

## ΣΥΝΤΟΜΟΓΡΑΦΙΕΣ

**WWW** World Wide Web

**HTML** HyperText Markup Language

**CSS**Cascading Style Sheets

**SQL**Structured Query Language

**UML**Unified Modeling Language

**CSV**Comma Separated Values

**GaussianNB** Gaussian Naive Bayes

**SVM** Support Vector Machine



### ΕΙΣΑΓΩΓΗ

Αντικείμενο της παρούσας πτυχιακής εργασίας είναι η ανάπτυξη μίας εφαρμογής σε μορφή ιστοσελίδας στην οποία εξουσιοδοτημένοι ιατροί θα μπορούν να κάνουν αναζήτηση στη βάση δεδομένων για τον ασθενή και να δουν το ιστορικό του να του χορηγήσουν φάρμακα και να σημειώσουν την διάγνωση και συμπεράσματα καθώς και να κάνουν μια πρόβλεψη αν έχει διαβήτη συμπληρώνοντας μια φόρμα με παραμέτρους. Επίσης ο ασθενείς θα έχει την δυνατότητα να μπαίνει στην ειδικά διαμορφωμένη ιστοσελίδα για αυτούς όπου θα βλέπει το ιστορικό του και θα κλείνει ραντεβού.

## Ιατρικός φάκελος με δυνατότητα πρόβλεψης διαβήτη



## **ΚΕΦΑΛΑΙΟ 1**

### **Διαδίκτυο και τεχνολογίες**

#### **ΕΙΣΑΓΩΓΗ**

Στο κεφάλαιο αυτό θα παρουσιαστούν ορισμένες δικτυακές τεχνολογίες όπως το θεμελιώδες μοντέλο TCP/IP και πρωτοκολλά καθώς και ένα κομμάτι της ιστορίας του παγκόσμιου ιστού

#### **1.1 Ιστορική αναδρομή**

Το 1969 ιδρύθηκε το πρώτο δίκτυο μεταγωγής πακέτου ARPANET (Advanced Research Projects Agency Network) στην Αμερική από το Γραφείο ερευνών Αμύνης DARPA (Defense Advanced Research Projects Agency) για χρήση στα πανεπιστήμια και εργαστήρια ερευνών. Για την διασύνδεση δυο υπολογιστών χρησιμοποιήθηκε το πρωτόκολλο NCP (Network Control Protocol) το οποίο αντικαταστάθηκε από το TCP (Transmission Control Protocol). Μάζι με το ARPANET διμοιθουργιθικαν και άλλα δίκτυα το στρατιωτικό MILNET το NLP (National Physical Laboratory) για εμπορικούς λόγους το CYCLADES στη Γαλλία για επιστημονικούς λόγους και το ALOHA στη Χαβάη. Για να αποτρέψουν την συμφόρηση γραμμών μέσου στο NPL δικτύου λόγω αυξημένης κίνησης δημιουργήθηκε το packet switching το οποίο σπάει το πακέτο σε μικρότερα τμήματα και τα μεταφέρει στον παραλήπτη όπου εκεί συναρμολογείτε όλο το πακέτο. Τα δίκτυα αυτά οδήγησαν στην δημιουργία του internet. Το 1989 ο Άγγλος επιστήμονας Tim Berners-Lee εφηύρε το παγκόσμιο ιστό (WWW) το

## **Ιατρικός φάκελος με δυνατότητα πρόβλεψης διαβήτη**

οποίο είναι ένα σύνολο δικτύων. Ο παγκόσμιος ιστός περιέχει όλη την πληροφορία και καθένας έχει την πρόσβαση σε αυτόν. Μάζι με τον παγκόσμιο ιστό αναπτύχθηκε το URL (Uniform Resource Locator) και το πρωτόκολλο HTTP (Hypertext Transfer Protocol).

### **1.2 Μοντέλο TCP/IP**

Το TCP/IP είναι μια αρχιτεκτονική για να διασύνδεση άλλες δικτυακές συσκευές όπως είναι τα δορυφορικά δίκτυα. Η αρχιτεκτονική αυτή απαρτίζεται από τέσσερα επίπεδα. Επίπεδο εφαρμογών το οποίο παρέχει πρωτόκολλα όπως HTTP και DNS, μεταφοράς (TCP και UDP), διαδικτύου (IP, ICMP) και συνδέσμου (DSL, Ethernet).

### **1.3 URL και HTTP**

Το πρωτόκολλο HTTP (Πρωτόκολλο μεταφοράς υπερκειμένου) χρησιμοποιείται για την μεταφορά δεδομένων ανάμεσα στον server και client. Επίσης για την ασφαλή μεταφορά δεδομένων χρησιμοποιείται το HTTPS. Το URL είναι μια διεύθυνση στο παγκόσμιο ιστό η οποία αναφέρετε σε πληροφορία η ιστοσελίδα την οποία και θα ανάκτηση στην οθόνη του πελάτη (client).

### **1.4 Πρωτόκολλο DNS**

Το Σύστημα Ονομάτων Περιοχών DNS (Domain Name Service) μας παρέχει έναν μηχανισμό αντιστοίχιση των IP διευθύνσεων με ονόματα όπως η διεύθυνση URL [www.teipir.gr](http://www.teipir.gr) αντιστοιχεί στην IP 195.251.92.20. Κάθε DNS server είναι υπεύθυνο για τα ονόματα που βρίσκονται εντός τις επικράτειας που διαχειρίζεται.

## Ιατρικός φάκελος με δυνατότητα πρόβλεψης διαβήτη

### 1.5 Πρωτόκολλο TCP και UDP

Τα πρωτόκολλα TCP και UDP βρίσκονται στο επίπεδο μεταφοράς του μοντέλου TCP/IP και παρέχει υπηρεσίες στους υπολογιστές προέλευσης και προορισμού να συνομιλούν.

Το Πρωτόκολλο Έλεγχου Μεταφοράς TCP (TransmissionControlProtocol) είναι ένα αξιόπιστο πρωτόκολλο δηλαδή τα πακέτα παραδίδονται χωρίς σφάλματα. Το TCP τεμαχίζει την εισερχόμενη ροή δεδομένων σε διακριτά μηνύματα και μεταβιβάζει το καθένα στο επίπεδο διαδικτύου. Στον προορισμό η διεργασία παράλυτης του TCP αποσυναρμολογεί τα μηνύματα που λαμβάνει.

Πρωτόκολλο Αυτοδύναμων Πακέτων Χρήστη UDP (UserDataProtocol) είναι ένα αναξιόπιστο πρωτόκολλο το οποίο χρησιμοποιείτε σε εφαρμογές όπου η γρήγορη παράδοση είναι πιο σημαντική από την ακριβή όπως μετάδοση ομιλίας και βίντεο. Επίσης σε αντίθεση με το TCP τα πακέτα του UDP δεν παραδίδονται με τη σωστή σειρά

### 1.6 Πρωτόκολλο IP

Το επίπεδο διαδικτύου στο μοντέλο TCP/IP μας παρέχει τα πρωτοκολλά IP (Internet Protocol) και ICMP (Internet Control Message Protocol) η δουλειά του επιπέδου αυτού είναι να παραδίδει τα πακέτα IP στο προορισμό τους. Η κύρια εφαρμογή του πρωτοκόλλου διαδικτύου IP είναι η στήριξη τις παγκόσμιας διασύνδεσης δηλαδή ο καθένας μπορούσε να συνδεθεί με οποιονδήποτε άλλον. Για να γίνει αυτό το πρωτόκολλο IP εφαρμόζει μια τεχνολογία δρομολογήσεις και διευθυνσιοδότησης. Για να φτάσου τα πακέτα στον προορισμό τους

χρησιμοποιούν την διεύθυνση IP (IP address) πχ (10.250.250.0) και προσδιορίζει τη σύνδεση μιας συσκευής στο δίκτυο η οποία για την έκδοση 4 (IPv4) έχει μήκος 32 bit. Οι διευθύνσεις IP κατατάσσονται σε τρεις κλάσεις A, B, και C

## Ιατρικός φάκελος με δυνατότητα πρόβλεψης διαβήτη

Κλάση (bits)	Δίκτυο	Υπολογιστής	Χρήση
A (0)	7 bit (0-127)	24 bit	Μεγάλα δίκτυα
B (10)	14 bit (128-191)	16 bit	Μεσαία δίκτυα
C (110)	21 bit (192-223)	8 bit	Μικρά δίκτυα

**Πίνακας 1.1** Κλάσεις IP διευθύνσεων

147	10	13	28	
10	010011	00001010	00001101	00011100
B	Δίκτυο	υπολογιστή		

**Πίνακας 1.2** Παράδειγμα IP διεύθυνσης 147.10.13.28

Η διεύθυνση ανήκει στην κλάση B γιατί ξεκινάει με τα bit (10).

Τα επόμενα 14bit προσδιορίζουν το δίκτυο το οποίο είναι το 147.10.

Τέλος τα τελευταία 16bit είναι για τον υπολογιστή .13.28

### 1.7 εξυπηρετητής (server)

Ο ρόλος των server είναι η εξυπηρέτηση αιτημάτων και παροχή διαφόρων υπηρεσιών στον πελάτη (client) ο οποίος είναι συνδεδεμένος με απομακρυσμένη σύνδεση παράδειγμα αιτήσεις προς τον server είναι ανακτήσει μιας ιστοσελίδας.

Ένας από τους πιο γνωστούς server είναι ο apache. Είναι server του παγκόσμιου ιστού και είναι συμβατός με διάφορες αρχιτεκτονικές λειτουργικών συστημάτων (windows, linux, MAC OS) και βάσεων δεδομένων (oracle, MySQL) πράγμα που τον κάνει πολλή διαδεδομένο. Καθένας μπορεί να έχει έναν apache server στον υπολογιστή του χωρίς την απαίτηση υλικού για server αρκεί να κατεβάσει το λογισμικό του (<https://httpd.apache.org/download.cgi>). Μετά την εγκατάσταση του προγράμματος ο χρήστης έχει έναν εικονικό server στον υπολογιστή του.

Υπάρχουν διάφορα εύκολα πακέτα εγκατάστασης του apache server τα οποία

## Ιατρικός φάκελος με δυνατότητα πρόβλεψης διαβήτη

περιέχουν βάση δεδομένων και γλώσσα διαδικτυακού προγραμματισμού (PHP). Μερικά τέτοια πακέτα είναι το WAMP, XAMPP και AMPPS.

## ΚΕΦΑΛΑΙΟ 2

### Εργαλεία και γλώσσες ανάπτυξης

#### Εισαγωγή

Στο κεφάλαιο αυτό θα παρουσιαστούν οι γλώσσες διαδικτυακού προγραμματισμού, μοντελοποίησης και διαχείρισης βάσεων δεδομένων. Επίσης θα αναφερθούν εργαλεία ανάπτυξης διαδικτυακών εφαρμογών. Η δομή του κεφαλαίου ξεκινάει με τις γλώσσες διαδικτυακού προγραμματισμού και ακολουθεί η διαχείριση βάσεων δεδομένων.

#### 2.1 HTML

Η HTML είναι μια γλώσσα μορφοποίησης υπερκειμένου (Hyper Text Markup Language) και όχι προγραμματισμού και είναι το θεμελιώδες δομικό στοιχείο μιας ιστοσελίδας στο παγκόσμιο ιστό. Το 1990 ο επιστήμονας υπολογιστών Tim Berners-Lee από το Cern της Γενεύης δημιούργησε την HTML. Η γλώσσα χρησιμοποιείται όχι μόνο για την μορφοποίηση κείμενου αλλά και για την ενσωμάτωση βίντεο και εικόνας καθώς και δημιουργώντας διαδραστικές ιστοσελίδες.

Το συντακτικό της γλώσσας απαρτίζεται από ένα αριθμό tags για την μορφοποίηση του κείμενου και τη δημιουργία συνδέσμων. Τα tags ξεκινάνε με το σύμβολο "<" και τελειώνουν με το ">". Μέσα σε αυτές της γωνιώδης αγκύλες περιλαμβάνετε το όνομα της ετικέτας και η ιδιότητα της. Η βασική δομή κάθε



## Ιατρικός φάκελος με δυνατότητα πρόβλεψης διαβήτη

HTML αρχείου είναι `<html><head></head><body></body></html>`. Χάρης την ιδιαιτερότητα τις HTML να ενθέτη κώδικα PHP κάνοντας την ιστοσελίδα πιο λειτουργική και δυναμική. Αρχείο που έχει κωδικά HTML και PHP έχει την κατάληξη `.php` διαφορετικά αν έχει μονό κωδικά HTML θα έχει την καταλήξει `.html`.

```
<html>
  <head>
    <title>hello</title>
  </head>

  <body>
    <ul>
      <li>one</li>
      <li>two</li>
    </ul>
  </body>
</html>
```

Εικόνα 2.1 Παράδειγμα κωδικά HTML.

```
<html>
  <head>
    <title>example</title>
  </head>
  <body>
    <header>
      <h1><?php echo "hello world big";?></h1>
      <h2><?php echo "hello world small";?></h2>
    </header>
    <?php
      $sum1=1+2;
      $sum2=2+2;
    ?>
    <ul>
      <li><?php echo "sum1: ", $sum1;?></li>
      <li><?php echo "sum2: ", $sum2;?></li>
    </ul>
  </body>
</html>
```

Εικόνα 2.2 Παράδειγμα κωδικά HTML με PHP.

### 2.2 PHP

Η PHP είναι μια διαδομένη γλώσσα σεναριακού (script) διαδικτυακού προγραμματισμού και εκτελείτε από πλευράς εξυπηρετητή. Το συντακτικό τις PHP μοιάζει με τις C όμως ο κώδικας τις PHP δεν περνάει από μεταγλωττιστή\* όπως στην C αντίθετα ο κώδικας τις PHP εκτελείτε απευθείας από τον εκάστοτε διερμηνευτή\* διαβάζοντας βήμα βήμα τις τις εντολές και εκτελώντας τες.

## Ιατρικός φάκελος με δυνατότητα πρόβλεψης διαβήτη

Η χρήση της PHP μας παρέχει δυναμικές ιστοσελίδες πράγμα που την κάνει πρωτοπόρο στον διαδικτυακό προγραμματισμό και εντάσσεται σε πληθώρα εφαρμογών. Η συγγραφή του κωδικά της PHP μπορεί να γίνει σε έναν απλό κειμενογράφο (text editor) ή εργαλείο κειμενογράφου όπως το notepad++ και τα αρχεία πρέπει να έχουν επέκταση .php.

### 2.3 JavaScript

Η JavaScript είναι δυναμική γλώσσα προγραμματισμού και έχει συναρτήσεις ως αντικείμενα και είναι μια από τις μεγαλύτερες τεχνολογίες του παγκόσμιου ιστού όπου συμβάλει στις διαδραστικές ιστοσελίδες. Παραλού που η JavaScript δεν συσχετίζεται με την JAVA έχει πάρει όμως την θεμελιώδη αρχιτεκτονική δομή όπως το συντακτικό και τις πρότυπες (standard) βιβλιοθήκες. Η JavaScript δεν χρειάζεται μεταγλωττιστή για να εκτελεσθή αντί αυτού εκτελείται από μεριάς του πελάτη στον περιηγητή (browser) του.

Ο κώδικας τις JavaScript περικλείεται ανάμεσα σε δυο ετικέτες <script>κωδικας JavaScript</script>. Παρακάτω ακόλουθη ένα πρόγραμμα σε JavaScript το οποίο η συνάρτηση myfunction θα πρόσθεση δυο αριθμούς και θα επιστρέψει το άθροισμα τους.

```
<html>
<body>

<h2>JavaScript Functions</h2>

<p id="sum"></p>

<script>
function myFunction(p1, p2) {
    return p1 + p2;
}
document.getElementById("sum").innerHTML = myFunction(4, 3);
</script>

</body>
</html>
```

**Εικόνα 2.3** Παράδειγμα κωδικάjavascript

### 2.4 CSS

CSS(Cascading Style Sheets) είναι γλώσσα μορφοποιήσεων δηλαδή είναι υπεύθυνη για το πως θα είναι η παρουσίαση μιας ιστοσελίδας. Ο κώδικας τις CSS  
**Βασίλειος πεσκελίδης**

## Ιατρικός φάκελος με δυνατότητα πρόβλεψης διαβήτη

περικλείετε ανάμεσα σε δυο ετικέτες `<style>κωδικας CSS</style>` και μπορεί να εμφωλεύθη σε ένα αρχείο html ή σε ξεχωριστό αρχείο με επέκταση .css το οποίο η HTML θα καλέσει με κατάλληλο κώδικα για να συμπεριλάβει τις κλάσεις για την μορφοποίηση. Η CSS μας παρέχει πληθώρα εντολών για τη μορφοποίηση έτσι ώστε να φτιάξουμε μια εμφανίσιμη ιστοσελίδα στα μέτρα μας ή του πελάτη Μέχρι στιγμής κυκλοφορούν τρεις εκδόσεις CSS(CSS1, CSS2, CSS3) με τη πρώτη έκδοση η οποία καθορίστηκε το 1996.

### 2.5 MySQL

Η MySQL είναι λογισμικό ανοικτού κωδικά για τη διαχείριση βάσεων δεδομένων και δημιουργία σχεσιακών μοντέλων ή πινάκων στη βάση με τη γλώσσα SQL. Το ρεπερτόριο εντολών τις SQL μας παρέχει τόσο τη δυνατότητα δημιουργίας πινάκων στη βάση όσο και να τα προσπελάσουμε κάνοντας ερωτήματα (queries) προς τη βάση. Η MySQL τρέχει σε έναν εξυπηρετητή με αποτέλεσμα να παρέχει τη δυνατότητα πρόσβασης πολλών χρηστών σε ένα σύνολο βάσεων δεδομένων. Είναι συμβατή με πολλές αρχιτεκτονικές λειτουργικών συστημάτων και συνδυάζετε με αρκετές γλώσσες προγραμματισμού για τη μεταφορά δεδομένων από και προς τη βάση

Η διεπαφή προγραμματιστή και βάσης γίνεται είτε με τερματικό(κονσόλα) ή με ιδικό λογισμικό(Navicat, dbforg) για πιο εύχρηστη διαχείριση και σχεδιασμό.

### 2.6 Python

Είναι υψηλού επιπέδου αντικειμενοστρεφής γλώσσα προγραμματισμού. Σχεδιάστηκε έτσι ώστε να είναι ευανάγνωστη, εύχρηστη και η έννοιες να εκφράζονται με λιγότερες εντολές και γραμμές κώδικα Η χρήση της γίνεται με τη εγκατάσταση του διερμηνευτή python και είναι συμβατή με πολλά λειτουργικά Συστήματα. Επίσης μας παρέχει διάφορες βιβλιοθήκες όπως στατιστικής ανάλυσης, μηχανικής μάθησης και βιοπληροφορικής.

### 2.7 UML

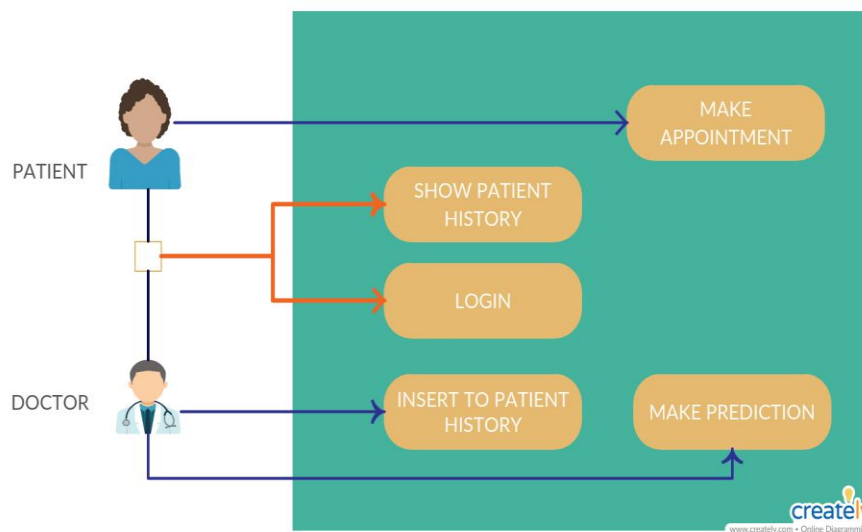
Η γλώσσα μοντελοποίησης (Unified Model Language) χρησιμοποιείται για την γραφική απεικόνιση και περιγραφή λογισμικού συστήματος. Συνήθως χρησιμοποιείται στη μηχανική λογισμικού για να γίνει κατανοητό το λογισμικό σχέδιο από τους προγραμματιστές και τον πελάτη

## ΚΕΦΑΛΑΙΟ 3

### Ιατρικός φάκελος

#### Εισαγωγή

Στο παρόν κεφάλαιο θα παρουσιαστεί και θα αναλυθεί η διαδικτυακή εφαρμογή μαζί με την βάση δεδομένων. Συγκεκριμένα η αναφορά θα είναι τόσο για την ιστοσελίδα του ιατρού όσο και του ασθενή μαζί με τον κωδικά. Παρακάτω παρουσιάζετε το UML διάγραμμα περιπτώσεων χρήσης της εφαρμογής.



**Εικόνα 3.1** UML διάγραμμα περιπτώσεων χρήσης

Το σύστημα έχει την δυνατότητα να μην επιτρέπει την πρόσβαση σε μη δικαιούχους, προστατεύοντας έτσι, το ιατρικό απόρρητο. Προορίζεται για να χρησιμοποιηθεί από νοσοκομεία, διαγνωστικά κέντρα και λοιπά νοσηλευτικά

## Ιατρικός φάκελος με δυνατότητα πρόβλεψης διαβήτη

ιδρύματα. Απευθύνεται σε ασθενείς, ιατρικό και νοσηλευτικό προσωπικό, καθώς επίσης και στο ανθρώπινο δυναμικό της γραμματείας.

Σε κάθε εξουσιοδοτημένο χρήστη του δίνετε ο κωδικός και το όνομα χρήστη γι να μπορέσει να μπει στην εφαρμογή. Επίσης, ανά μήνα το σύστημα ζητάει από τους Ιατρούς να κάνουν αλλαγή τους κωδικούς τους, ενώ για τους ασθενείς είναι προορατική η αλλαγή

### Επιχειρησιακοί κανόνες

- Η Γραμματεία έχει δικαιοδοσία :
  - i. Να δημιουργεί έναν νέο ιατρικό φάκελο.
  - ii. Να εισάγει νέα στοιχεία στον ιατρικό φάκελο.
  - iii. Να αναζητά στο ιστορικό του ιατρικού φακέλου.
  - iv. Να βλέπει τον ιατρικό φάκελο.
  - v. Να αλλάζει τα στοιχεία του ασθενούς σε έναν ιατρικό φάκελο.
  
- Οι Γιατροί επιτρέπεται:
  - i. Να βλέπουν τον ιατρικό φάκελο του ασθενούς.
  - ii. Να κάνουν αναζήτηση στο ιστορικό του ιατρικού φακέλου.

### Γενική περιγραφή των λειτουργιών

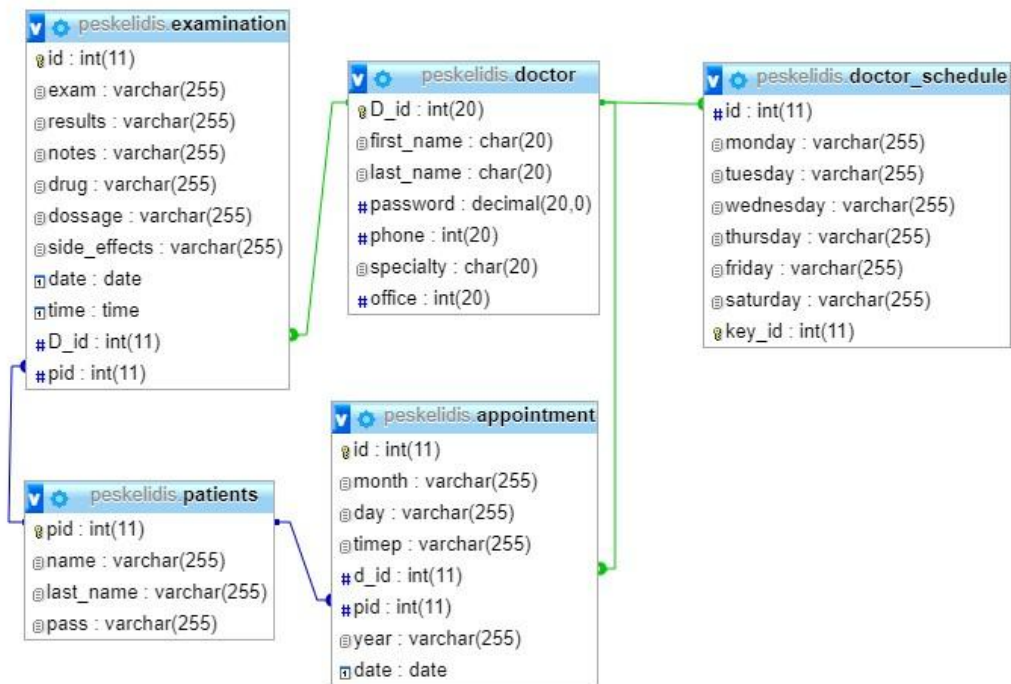
- Δημιουργία ιατρικού φακέλου ασθενούς
- Επεξεργασία ιατρικού φακέλου ασθενούς
- Διαγραφή ιατρικού φακέλου ασθενούς
- Εισαγωγή στοιχείων στον ιατρικό φάκελο ασθενούς
- Επεξεργασία στοιχείων του ιατρικού φακέλου ασθενούς
- Διαγραφή στοιχείων από τον ιατρικό φάκελο ασθενούς
- Αναζήτηση ιατρικού φακέλου ασθενούς
- Αναζήτηση στοιχείων στον ιατρικό φάκελο ασθενούς

# Ιατρικός φάκελος με δυνατότητα πρόβλεψης διαβήτη

## 3.1 Βαση δεδομένων

Η βάση δεδομένων απαρτίζεται από πέντε πίνακες και βασίζεται στην σχεσιακό μοντέλο της MySQL και είναι σχεδιασμένη με την γλώσσα SQL. Το πρωτεύον κλειδί (primary key) σε έναν πίνακα είναι ένα πεδίο με τη δυνατότητα να εξασφαλίζει ότι η εγγραφή είναι μοναδική. Ενώ το ξένο κλειδί (foreign key) μας παρέχει τη δυνατότητα να συσχετίσουμε άλλο πεδίο ενός άλλου πίνακα

Παρακάτω παρουσιάζετε το σχεσιακό μοντέλο της βάσης δεδομένων. Η γραμμές προσδιορίζουν την συσχέτιση μεταξύ των πινάκων με την έντονη κουκκίδα στο τέλος κάθε γραμμής να είναι το ξένο κλειδί



Εικόνα 3.2 Σχεσιακό μοντέλο της βάσης

Ο πίνακας examination περιέχει το ιστορικό του ασθενή, αφού στον πίνακα αυτόν γίνεται καταγραφή των στοιχείων της εξέτασης όπως το είδος της εξέτασης, χορήγηση φαρμάκου και η ημερομηνία. Επίσης έχει δυο ξένα κλειδιά το πεδίο pid (patient id) ο οποίος είναι αριθμός του ασθενή και είναι πρωτεύον κλειδί στο

## Ιατρικός φάκελος με δυνατότητα πρόβλεψης διαβήτη

πινάκα patients και το πεδίο D\_id (doctor id) που είναι το πρωτεύον κλειδί στον πίνακα doctor. Στον πίνακα examination έχουν δικαίωμα να δουν τα στοιχεία του ο ασθενής, ιατρός και η γραμματεία αλλά μόνο ο Ιατρός έχει το δικαίωμα να γράψει στον πίνακα. Ενώ η οποιαδήποτε διαγραφή δεδομένων από οποιονδήποτε πίνακα γίνεται μόνο από την γραμματεία.

Τα ραντεβού κάθε ασθενή καταγράφονται στον πίνακα appointment ο οποίος πίνακας περιέχει τα βασικά πεδία για ραντεβού όπως μήνας, ημέρα, ώρα. Τα δεδομένα του ασθενή και Ιατρού σχετίζονται με τον πίνακα appointment μέσω των ξένων κλειδιών pid (patientid) και d\_id (doctorid) τα οποία τα πεδία αυτά είναι πρωτεύον κλειδί στον αντίστοιχο πίνακα patients και doctor. Στον πίνακα appointment μόνο ο ασθενής έχει το δικαίωμα να κάνει εγγραφή δεδομένων στον πίνακα ενώ η γραμματεία έχει μόνο το δικαίωμα να κάνει διαγραφή. Επίσης ο Ιατρός μπορεί να δει τα δικά του ραντεβού μόνο για την τρέχουσα ημέρα και αν υπάρχουν.

Για να εμφανίσουμε τα δεδομένα του πίνακα examination μαζί με τα πεδία του πίνακα patient και doctor χρησιμοποιούμε την λειτουργία INNER JOIN τις SQL. Η λειτουργία αυτή συνδυάζει στήλες ή εγγραφές από έναν ή περισσότερους πίνακες. Για να εφαρμοστεί η λειτουργία JOIN πρέπει να υπάρχει συσχέτιση μεταξύ των πινάκων δηλαδή η δομή πρωτεύον και ξένο κλειδί

```
SELECT column-names
FROM table-name1 INNER JOIN table-name2
ON column-name1 = column-name2
WHERE condition
```

**Εικόνα 3.3** Σύνταξη της INNER JOIN.

Η σύνταξη ξεκινάει επιλέγοντας τα πεδία που θέλουμε από τους πίνακες (column-names) στη συνέχεια αναφέρουμε το όνομα του πίνακα με το πρωτεύον κλειδί (table-name1) ακολουθούμενο από την λειτουργία INNER JOIN αναφέροντας τον δεύτερο πίνακα (table-name2). Τέλος, γράφουμε τα πεδία που συσχετίζουν τους



## Ιατρικός φάκελος με δυνατότητα πρόβλεψης διαβήτη

δύο πίνακες (column-name1, column-name2), πρωτεύον και ξένο κλειδί αντίστοιχα.

Στη δική μας περίπτωση τα πεδία που έχουμε να ενώσουμε είναι από τρεις πίνακες οι οποίοι συσχετίζονται μεταξύ τους. Έτσι θα έχουμε δυο INNER JOIN και το αποτέλεσμα θα είναι ένας πίνακας με εγγραφές από τους πίνακες που έχουμε επιλέξει τα πεδία τους.

```
SELECT
    doctor.first_name,
    doctor.last_name,
    doctor.specialty,
    examination.exam,
    examination.results,
    examination.notes,
    examination.drug,
    examination.dosage,
    examination.side_effects,
    examination.date,
    examination.time,
    patients.`name`,
    patients.last_name,
    patients.pid
FROM
    doctor
INNER JOIN examination ON doctor.D_id = examination.D_id
INNER JOIN patients ON examination.pid = patients.pid
WHERE patients.pid =1
```

**Εικόνα.3.4**SQL κώδικας για το ιστορικό του ασθενή

```
SELECT
    appointment.day,
    appointment.`month`,
    appointment.timep,
    appointment.`year`,
    doctor.first_name,
    doctor.last_name,
    doctor.specialty
FROM
    appointment
INNER JOIN doctor ON appointment.d_id = doctor.D_id
where appointment.day >= '". $date.'" AND appointment.pid='". $pi.'" ;
```

**Εικόνα.3.5**SQL κώδικας για εμφάνιση προγραμματισμένων ραντεβού.

Για να συνδεθούμε και να έχουμε μεταφορά από και προς τη βάση χρησιμοποιούμε τη συνάρτηση mysqli. Η σύνδεση πραγματοποιείται γράφοντας new mysqli(host, user, password, database) το new στη PHP σημαίνει ότι δημιουργήσαμε ένα αντικείμενο mysqli. Με τη συνάρτηση αυτή επιταχύνουμε τι

## Ιατρικός φάκελος με δυνατότητα πρόβλεψης διαβήτη

διασύνδεση με τη βάση χρησιμοποιώντας τέσσερις παραμέτρους το όνομα του διακομιστή που έχει τη βάση, όνομα χρήστη, κωδικός και τέλος το όνομα της βάσης. Η συνάρτηση επιστρέφει ένα αντικείμενο το οποίο αναπαριστά τη σύνδεση με τη βάση και το εκχωρούμε σε μεταβλητή για να ελέγξουμε αν η σύνδεση έγινε σωστά Για κάθε ερώτημα που θα κάνουμε προς τη βάση θα χρησιμοποιούμε αυτή τη μεταβλητή για να καλέσουμε τη μέθοδο query(ερώτημα προς τη βάση) που ανήκει στο αντικείμενο mysqli.

```
$conn = new mysqli("localhost:3306", "peskelidis", "", "peskelidis");  
    // Check connection  
    if ($conn->connect_error) {  
        die("Connection failed: " . $conn->connect_error);  
    }
```

**Εικόνα.3.6**Σύνδεση με τη βάση

```
$result = $conn->query($sql);
```

**Εικόνα.3.7**Αποτέλεσμα ερωτήματος

Η μεταβλητή \$result περιέχει το αποτέλεσμα του ερωτήματος δηλαδή τις έγγραφες. Για να τις προσπελάσουμε χρησιμοποιούμε τις συνάρτηση fetch\_assoc() για να ανακτήσουμε μια εγγραφή και σε συνδυασμό με το βρόγχο while τις ανακτούμε όλες.

```
while($row = $result->fetch_assoc())
```

**Εικόνα.3.8**Προσπέλαση εγγραφών.

### 3.2 Ιστοσελίδα Ιατρού

Η Ιατρική φόρμα αποτελείται από τέσσερα αρχεία δυο με κατάληξη .php και τα άλλα με .css. Τα δυο πρώτα αρχεία με κατάληξη .php αποτελούν την φόρμα εισαγωγής στοιχείων (login.php) και την βασική φόρμα διαχείρισης Ιατρού (doctor.php). Τα άλλα δυο έχουν να κάνουν με την μορφοποίηση τις ιστοσελίδας (login.css, doctor.css). Επίσης για το δίγλωσσο μενού υπάρχουν δυο αρχεία το GR.php και το UK.php τα οποία περιλαμβάνουν τις λέξεις του μενού το ένα στα ελληνικά και το άλλο στα αγγλικά.

## Ιατρικός φάκελος με δυνατότητα πρόβλεψης διαβήτη

### 3.2.1 Φόρμα εισόδου

Στη φόρμα εισόδου ο Ιατρός πρέπει να βάλει το όνομα και το κωδικό του και ύστερα αφού πατήσει το κουμπί είσοδος γίνετε έλεγχος των στοιχείων που έβαλε Αν τα στοιχεία είναι σωστά τότε ο Ιατρός θα μεταβεί στην φόρμα διαχείρισης Ιατρού (doctor.php).



Εικόνα 3.6 Φόρμα εισόδου.

Η μορφοποίηση της φόρμας βασίζεται στη CSS και ο κώδικας της είναι σε ξεχωριστό αρχείο το οποίο έχει συμπεριληφθεί στην ετικέτα <link> της HTML. Τα πεδία και το κουμπί εισόδου της φόρμας είναι ομαδοποιημένα κάτω από την ετικέτα <div></div> και έχει μορφοποιηθεί με την CSS καλώντας την κλάση container η οποία αναφέρετε σε τμήμα κώδικα που βρίσκεται στο αρχείο CSS και είναι υπεύθυνο για την μορφοποίηση του πλαισίου. Αντίστοιχα η μορφοποίηση κάθε τμήματος μέσα στη φόρμα θα γίνεται αναφέροντας την κλάση ακολουθούμενη από το όνομα της ετικέτας που θέλουμε τροποποιήσουμε.

```
<link rel="stylesheet" type="text/css" href="doctor_login_style.css"></link>
<div class="container">

  <form action="" method="get">
    <h1><i> <?=$login;?> </i></h1>

    <div class="form-input">
      <input type="text" name="fname" placeholder= <?=$fname;?> >
      <input type="password" name="pass" placeholder= <?=$pass;?> >
    </div>
    <button onclick='submit'><b><?=$search;?></b></button>

  </form>
</div>
```

## Ιατρικός φάκελος με δυνατότητα πρόβλεψης διαβήτη

Εικ 3.7 Κώδικας φόρμας εισόδου.

```
body{
margin: 0 auto;
background-image: url("cal.jpg");
background-repeat: no-repeat;
background-size: 100% 650px;
}

.container{
width: 350px;
height: 200px;
text-align: center;
margin: 0 auto;
background-color: rgba(52, 73, 94, 0.2);
border-radius: 5px;
margin: 0 auto;
margin-top: 90px;
}

.container input{
height: 30px;
width: 200px;
opacity: 0.7;
filter: alpha(opacity=50);
margin-top: 2;
background-color: #fff;
padding-left: 15px;
border-radius: 2px;
border-color: rgba(255, 230, 230, 0.7);
}

.container h1{
opacity: 0.7;
filter: alpha(opacity=50);
}

.container button{
margin-top: 20px;
height: 30px;
width: 80px;
background-color: rgba(222, 239, 243, 0.9);
border-color: rgba(255, 230, 230, 0.8);
border-radius: 4px;
}
```

Εικ 3.8 Κώδικας CSS.

Ο Ιατρός έχει τη δυνατότητα να αλλάξει τη γλώσσα του μενού επιλέγοντας μια από τις δυο επιλογές στην επάνω αριστερή άκρη της οθόνης. Οι δυο επιλογές είναι radio buttons και κάθε φορά που πατηθεί ένα από τα δυο η html στέλνει την τιμή value μέσω της μεθόδου POST στην PHP και ανάλογα με την τιμή τις value η PHP φορτώνει το αντίστοιχο αρχείο με τους χαρακτήρες τις εκάστοτε γλώσσας. Η εντολή που συμπεριλαμβάνει το αρχείο χαρακτήρων είναι η συνάρτηση include ακολουθούμενη από το όνομα του αρχείου. Η συνάρτηση αυτή παίρνει τον κώδικα από το αρχείο που καλούμε και τον αντιγράφει στο αρχείο με τη συνάρτηση include.

```
<?php//radio button language?>
<form action="" method="post">
<input type="radio" name="language" onclick="javascript: submit()" value="english">
English
<input type="radio" name="language" onclick="javascript: submit()" value="greek">
Ελληνικά
</form>

<?php
if(isset($_POST['language']))
    $sub=$_POST['language'];
?>
<?php//end radio button?>

<?php
if ($sub == 'english')
include 'menuUK.php';
else if ($sub == 'greek')
include 'menuGR.php';
?>
```

## Ιατρικός φάκελος με δυνατότητα πρόβλεψης διαβήτη

**Εικόνα 3.9**Κώδικας επιλογής γλώσσας και εξωτερικού αρχείου.

Από τη στιγμή που πατήσει ο Ιατρός είσοδος τα δεδομένα που πληκτρολόγησε στη φόρμα θα μεταβούν στην PHP μέσω της μεθόδου POST. Ύστερα γίνεται ο έλεγχος των δεδομένων που έβαλε ο Ιατρός με αυτά της βάσης δεδομένων. Αν τα στοιχεία είναι σωστά θα η PHP θα πάρει τα στοιχεία του Ιατρού από την βάση και θα τα περάσει στην επομένη φόρμα που θα μεταβεί ο Ιατρός.

Η POST ανήκει στη οικογένεια μεθόδων του πρωτοκόλλου HTTP (Hypertext Transfer Protocol) το οποίο έχει σχεδιαστεί για να επιτρέπει την επικοινωνία μεταξύ πελατη-εξυπηρετητή και διακομιστών και λειτουργεί ως πρωτόκολλο αίτησης-απόκρισης μεταξύ ενός πελάτη και ενός διακομιστή. Η μέθοδος POST χρησιμοποιείται για να στείλει δεδομένα μέσω του HTTP αιτήματος στον εξυπηρετητή. Επίσης είναι ασφαλές γιατί τα δεδομένα δεν αποθηκεύονται στον περιηγητή και δεν είναι ορατά στην URL.

```
$sql = "SELECT first_name,password, last_name, specialty, office FROM doctor";
$result = $conn->query($sql);

if ($result->num_rows > 0) {
    // output data of each row
    while($row = $result->fetch_assoc()) {
        $f_name=$row["first_name"];
        $l_name=$row["last_name"];
        $specialty=$row["specialty"];
        $office=$row["office"];
        $pas=$row["password"];

        if(isset($f_name) == isset($name) && $pas == $pass){
            $_SESSION['lang'] = "$sub";
            $_SESSION['doc_name'] = "$f_name";
            $_SESSION['doc_lname'] = "$l_name";
            $_SESSION['doc_specialty'] = "$specialty";
            $_SESSION['doc_office'] = "$office";
            header('Location:doctor_page.php');
        }
    }
}
```

**Εικόνα 3.10**Κώδικας ελέγχου δεδομένων.

Για να γίνει ο έλεγχος δεδομένων και η ταυτοποίηση του χρήστη πρώτα κάνουμε ένα ερώτημα προς τη βάση για να πάρουμε τα δεδομένα του Ιατρού και τον κωδικό του από τον πίνακα doctor και το αποθηκεύουμε στη μεταβλητή \$sql. Στη συνέχεια καλούμε τη συνάρτηση query() με παράμετρο το ερώτημα προς τη βάση η οποία θα μας επιστρέψει τα δεδομένα σε διαφορετικές μορφές. Μέσα στον βρόγχο αποθηκεύουμε τα δεδομένα του Ιατρού σε μεταβλητές για να τα

## Ιατρικός φάκελος με δυνατότητα πρόβλεψης διαβήτη

περάσουμε στην επομένη σελίδα μέσω `session()`. Τέλος συγκρίνουμε με την `if` αν το όνομα και ο κωδικός που έβαλε ο χρήστης είναι σωστά τότε περνάμε τα δεδομένα στο `session` και καλούμε την επόμενη φόρμα με την εντολή `header()` η οποία περιέχει το όνομα του αρχείου.

Οι συνεδρίες (`sessions`) είναι ένας εναλλακτικός τρόπος να κάνουμε τα δεδομένα προσπελάσιμα από όλες τις σελίδες που βρίσκονται στον εξυπηρετητή. Τα `sessions` δημιουργούν ένα αρχείο στον κατάλογο του εξυπηρετητή όπου και εκεί αποθηκεύονται η μεταβλητές και είναι προσπελάσιμες από διαφορετικές σελίδες. Για να δημιουργήσουμε ένα `session` καλούμε τη συνάρτηση `session_start()` και η μεταβλητές αποθηκεύονται σε έναν υπερκαθολικό πίνακα `$_SESSION[ ]`. Επίσης χρησιμοποιούμε τη συνάρτηση `isset()` για να ελέγξουμε εάν η συνεδριακή μεταβλητή έχει οριστεί ή όχι

### 3.2.2 Ιατρική σελίδα

Αφού ο Ιατρός βάλει σωστά δεδομένα στη φόρμα εισαγωγείς τότε θα μεταβεί στη δευτέρα Ιατρική φόρμα όπου εκεί θα βλέπει το ιστορικό του ασθενή, θα χορηγεί φάρμακα και τέλος θα κάνει πρόβλεψη για διαβήτη. Η φόρμα περιλαμβάνει μενού επιλογής γλώσσας πεδίο άσογης AMKA ασθενή και κουμπί ύπουλης καθώς και τρία κουμπιά που εμφανίζουν το ιστορικό, εξέταση και πρόβλεψη. Πατώντας ένα από τα τρία κουμπιά θα εμφανιστή η αντίστοιχη επιλογή. Η λειτουργία αυτή περιλαμβάνει τη μέθοδο `getElementById` τις `javascript`. Συγκεκριμένα, το πλαίσιο που εμφανίζετε πατώντας καθένα από τα τρία κουμπιά βρίσκετε μέσα σε ένα `<div>`. Η ετικέτα αυτή ομαδοποιεί ένα τμήμα κώδικα με αποτέλεσμα να μπορούμε να του αναφερόμαστε με βάση το ID ή CLASS που το βαφτίσαμε. Έτσι πατώντας το κουμπί ΙΣΤΟΡΙΚΟ καλούμε τη συνάρτηση `myFunction_history()` η οποία με τη σειρά τις θα πάρει το `id` του `div` που περιέχει το ομαδοποιημένο κωδικά που εμφανίζει τι ιστορικό του ασθενή

```
<script>
function myFunction_history() {
    document.getElementById("myDropdown_history").classList.toggle("show");
}
</script>
```

Εικόνα 3.11 Κώδικας εμφάνισης μενού κουμπιού.

## Ιατρικός φάκελος με δυνατότητα πρόβλεψης διαβήτη

Στη δεύτερη φόρμα όπως και στη πρώτη υπάρχει η λειτουργία αλλαγής γλώσσας και ο Ιατρός μπορεί να επιλέξει ανάμεσα σε ελληνικό η αγγλικό μενού Όμως, μόλις μεταβεί ο Ιατρός στη δεύτερη φόρμα το μενού θα είναι στη ίδια γλώσσα που επέλεξε στη φόρμα εισαγωγείς. Αυτή τη λειτουργία την πετυχαίνουμε αποθηκεύοντας την επιλογή του radio button σε μια μεταβλητή και την περνάμε στη επόμενη φόρμα μέσω SESSION. Ύστερα γίνεται έλεγχος τις μεταβλητής με if και φορτώνετε το εκάστοτε αρχείο με τους χαρακτήρες τις αντίστοιχης γλώσσας.

Μαζί με την επιλογή του radio button στη δεύτερη φόρμα περνάνε τα δεδομένα του Ιατρού τα οποία ο Ιατρός μπορεί να τα δει φέρνοντας το ποντικί πάνω στην εικόνα επάνω δεξιά στην οθόνη όπου και θα εμφανιστεί ένας πίνακας με τα στοιχεία του Ιατρού.



ΟΝΟΜΑ	VASILIS
ΕΠΙΘΕΤΟ	PESKELIDIS
ΕΙΔΙΚΟΤΗΤΑ	cardiologist
ΓΡΑΦΕΙΟ	306

**Εικόνα 3.12** Δεδομένα Ιατρού.

Τα δεδομένα του Ιατρού εμφανίζονται μέσα σε HTML πίνακα και περικλείεται μέσα σε ετικέτα `<div></div>` η οποία περιέχει κλάση `dropdown3`. Η κλάση αυτή αναφέρετε σε μορφοποίηση του πλαισίου που περιέχει τον πίνακα HTML. Ο κώδικας CSS περικλείετε ανάμεσα σε ετικέτα `<style></style>` και για κάθε μέθοδο που χρησιμοποιούμε να μορφοποιήσουμε το πλαίσιο χρησιμοποιούμε την κλάση `dropdown3` ακολουθούμενη από το τμήμα που θέλουμε να μορφοποιήσουμε. Στον πίνακα οι χαρακτήρες τις αριστερής στήλης φορτώνονται από εξωτερικό αρχείο ανάλογα με την γλώσσα που έχει επιλέξει ο Ιατρός. Ενώ τα δεδομένα τις αριστερής στήλης έχουν από τη πρώτη φόρμα στη δεύτερη μέσω SESSION.

## Ιατρικός φάκελος με δυνατότητα πρόβλεψης διαβήτη

```
<style>
.dropdown3 {
  position: absolute;
  right:60;
  display: inline-block;
}

.dropdown3-content {
  display: none;
  position: absolute;
  right:7;
  background-color: #f9f9f9;
  min-width: 160px;
  box-shadow: 0px 0px 16px 0px rgba(0,0,0,0.2);
  z-index: 1;
}

.dropdown3:hover .dropdown3-content {
  display: block;
}

.desc {
  padding: 15px;
  text-align: center;
}
</style>
```

**Εικόνα 3.13**Κώδικας μορφοποίησης πλαισίου.

Ο κώδικας μορφοποίησης βρίσκεται στο ίδιο αρχείο με το κώδικα του πίνακα Μέσα στη κλάση `.dropdown3` στην ετικέτα `<style>` ορίζουμε το μέγεθος και τη θέση τις εικόνας που θα περιέχει το πλαίσιο και τον πίνακα Ενώ η γραμμή `.dropdown3-content` περιέχει τον κώδικα για τη μορφοποίηση του πλαισίου. Τέλος το `.dropdown3:hover` εμφανίζει το πλαίσιο μαζί με τον πίνακα με την εντολή `display:block` όταν περάσει το ποντίκι πάνω από την εικόνα

Για να δει ο Ιατρός το ιστορικό του ασθενή πρέπει να βάλει το ΑΜΚΑ του ασθενή στο περιεχόμενο αναζήτησης και να πατήσει υποβολή.

**Εικόνα 3.14**Περιεχόμενο αναζήτησης.

Το περιεχόμενο αναζήτησης είναι μια ετικέτα `<input>` τύπου `text` και το κείμενο μέσα είναι `placeholder="search a patient"`. Ενώ το κουμπί υποβολής είναι ετικέτα `<input>` τύπου `submit` και η τιμή του αλλάζει με το ποια γλώσσα έχει επιλεγθεί. Και οι δυο ετικέτες βρίσκονται μέσα σε μια ετικέτα `<form></form>` και για να πάρει η PHP το ΑΜΚΑ του ασθενή η ετικέτα `<form>` έχει μέθοδο `POST`.

```
<form action="" method="POST">
<input type="text" name="psearch" placeholder="search a patient">
<input type="submit" value=?=$submit;?>
</form>
```

**Εικόνα 3.15**Κώδικας περιεχομένου αναζήτησης



## Ιατρικός φάκελος με δυνατότητα πρόβλεψης διαβήτη

Στη συνέχεια για να δει ο Ιατρός το ιστορικό του ασθενή πρέπει να πατήσει το κουμπί με το όνομα ΙΣΤΟΡΙΚΟ και θα εμφανιστεί ένας πίνακας HTML με το ιστορικό του εκάστοτε ασθενή μαζί με τα ονόματα των συλών

ΙΣΤΟΡΙΚΟ	ΠΡΟΒΛΕΨΗ	ΕΞΕΤΑΣΗ									
ΟΝΟΜΑ_ΑΣΘΕΝΗ	ΕΠΙΘΕΤΟ_ΑΣΘΕΝΗ	ΗΜΕΡΟΜΗΝΙΑ	ΕΞΕΤΑΣΗ	ΔΙΑΓΝΩΣΗ	ΦΑΡΜΑΚΟ	ΔΟΣΟΛΟΓΙΑ	ΟΝΟΜΑ_ΓΙΑΤΡΟΥ	ΕΠΙΘΕΤΟ_ΓΙΑΤΡΟΥ	ΕΙΔΙΚΟΤΗΤΑ	ΗΜΕΡΟΜΗΝΙΑ	
ΝΙΚΟΣ	ΣΟΥΛΙΣ	2018-04-23	kardia	thromvosi	aspirini	1	VASILIS	PESKELIDIS	cardiologist	20:23:36	



Εικόνα 3.16 Πίνακας ιστορικού ασθενή.

Προκειμένου να εμφανίσουμε το ιστορικό πρέπει να κάνουμε το αντίστοιχο ερώτημα προς τη βάση. Πρέπει να επιλέξουμε τα δεδομένα του ασθενή, Ιατρού και τα πεδία της εξέτασης από τους αντίστοιχους πίνακες. Στη συνέχεια από τη στιγμή που οι τρεις πίνακες (doctor, patient και examination) συσχετίζονται μεταξύ τους εφαρμόζουμε δυο inner join για να ενώσουμε τα πεδία που επιλέξαμε από τους πίνακες και να πάρουμε το αποτέλεσμα. Επίσης χρησιμοποιούμε where για να επιλέξουμε τον ασθενή που θέλουμε μέσω του ΑΜΚΑ.

```
$sqlq="SELECT
doctor.first_name,
doctor.sur_name,
doctor.specialty,
examination.exam,
examination.results,
examination.notes,
examination.drug,
examination.dossage,
examination.side_effects,
examination.date,
examination.time,
patients.name,
patients.last_name,
patients.pid
FROM doctor
INNER JOIN examination ON doctor.D_id = examination.D_id
INNER JOIN patients ON examination.pid = patients.pid
WHERE patients.pid = ". $psearch. " ";
$result = $conn->query($sqlq);
while($row = $result->fetch_assoc()) {
    echo"<tr>
    <td>" $row["name"]. "</td>
    <td>" $row["last_name"]. "</td>
    <td>" $row["date"]. "</td>
    <td>" $row["exam"]. "</td>
    <td>" $row["results"]. "</td>
    <td>" $row["drug"]. "</td>
    <td>" $row["dossage"]. "</td>
    <td>" $row["first_name"]. "</td>
    <td>" $row["sur_name"]. "</td>
    <td>" $row["specialty"]. "</td>
    <td>" $row["time"]. "</td>
    </tr>";
```

Εικόνα 3.17 Κώδικας ιστορικού ασθενή.

Για να τα εμφανίσουμε σε πίνακα HTML πρέπει να εμφωλεύσουμε την HTML στην PHP αυτό το κάνουμε χρησιμοποιώντας τα δίκλα αυτάκια (") και μέσα σε αυτά

## Ιατρικός φάκελος με δυνατότητα πρόβλεψης διαβήτη

γράφουμε κώδικα HTML και έξω γράφουμε κώδικα PHP. Διαφορετικά χωρίς τα αυτάκια η PHP θα εμφάνιση συντακτικό σφάλμα

Πατώντας το κουμπί ΕΞΕΤΑΣΗ εμφανίζετε μια φόρμα στην οποία ο Ιατρός γράφει τα αποτελέσματα τις εξέτασης και χορηγεί φάρμακα στον ασθενή. Ύστερα αφού έχει συμπλήρωση τα πεδία πατάει το κουμπί υποβολή για να περαστούν τα δεδομένα στη βάση Η φόρμα αυτή απαρτίζεται από δυο HTML πίνακες και ένα κουμπί υποβολής. Στον πρώτο πίνακα καταχωρείτε η εξέταση που έγινε, η διάγνωση και ορισμένα σχόλια. Ενώ στον δεύτερο καταχωρεί το φάρμακο μαζί με τη διάρκεια χρήσης τις παρενέργειες και τη δοσολογία του.

Τα δεδομένα που εισήγαγε ο Ιατρός στη φόρμα θα αποθηκευτούν στον πίνακα examination μαζί με την τρέχουσα ημερομηνία που μας δίνει η συνάρτηση `date("d/m/Y")` τις PHP. Η καταχώριση των δεδομένων στη βάση γίνεται με ερώτημα προς τη βάση με την εντολή `INSERT()` ακολουθούμενη από το όνομα του πίνακα μαζί με τα πεδία που θέλουμε να κάνουμε τη καταχώριση και την εντολή `VALUES()` η οποία θα περιέχει τα δεδομένα. Επίσης το ερώτημα καταχώρισης πληροφορίας βρίσκετε σε ξεχωριστό αρχείο, έτσι τα δεδομένα αποστέλλονται στο ξεχωριστό αρχείο μέσω `POST` όπου και γίνεται η ανάθεση τους σε μεταβλητές.

```
mysqli_query($conn, "INSERT INTO examination(exam, results, notes, drug, dossage, side_effects, `date`, `time`, D_id, pid) VALUES ('$exam', '$diag', '$comm', '$drugg', '$doss', '$sidef', CURDATE(), CURTIME(), '$d_id.', '$AMKA.'");
```

**Εικόνα 3.18** Ερώτημα για την αποθήκευση δεδομένων τις εξέτασης.

Στην επιλογή ΠΡΟΒΛΕΨΗ εμφανίζεται μια φόρμα και περιέχει μια ετικέτα `<input>` στην οποία εισάγονται τα δεδομένα προς πρόβλεψη και μια `<select>` με τους τρεις αλγόριθμους και τέλος ένα κουμπί υποβολής. Στην ετικέτα `input` ο Ιατρός πρέπει να εισάγει τα δεδομένα με σωστό τρόπο έτσι όπως είναι υποδειγμένο μέσα στην ετικέτα (x,y,z). Στη συνέχεια επιλέγει έναν από τους τρεις αλγόριθμους (Gaussian Naive Bayes, Support Vector Machine, Random Forest Classifier) και πατάει υποβολή. Τα δεδομένα τις φόρμας και ο αλγόριθμος που επέλεξε ο Ιατρός μεταφέρονται μέσω `SESSION` σε ξεχωριστό αρχείο και εκεί αποθηκεύονται σε μεταβλητές και καλούμε τη συνάρτηση `shell_exec()`. Η συνάρτηση αυτή εκτελεί μια εντολή μέσω φλοιού του λειτουργικού συστήματος και επιστρέφει όλο το

## Ιατρικός φάκελος με δυνατότητα πρόβλεψης διαβήτη

αποτέλεσμα ως κορδόνι (string) σε αντίθεση με τη συνάρτηση `exec()` η οποία επιστρέφει μόνο τη τελευταία γραμμή του αποτελέσματος.

Η εντολή που παίρνει η συνάρτηση είναι ένα μονοπάτι προς τη `python` και το αντίστοιχο `python script` (`predict.py`) που θα εκτελέσει και θα κάνει τη πρόβλεψη. Μάζι με το μονοπάτι και το `script` περνάμε δυο ακόμα παραμέτρους τη `$datapredict` η οποία περιέχει τα δεδομένα για πρόβλεψη και ο αλγόριθμος που επέλεξε ο Ιατρός αποθηκεύετε στη δεύτερη παράμετρο τη `$pred_algorithm`. Οι παράμετροι αυτοί θα περαστούν ως ορίσματα στο `python script` μόλις το εκτελέσει ο φλοιός.

```
$flag=shell_exec("C:\Python27\python predict.py " $datapredict " " $pred_algorithm."");
```

**Εικόνα 3.19** Συνάρτησης `shell_exec()`.

Μέσο τις εντολής `import sys` στο `python script` επιτρέπουμε στις δυο παραμέτρους να περαστούν στη `python` ως ορίσματα και θα τα αποθηκεύσουμε σε μεταβλητές περνώντας το κάθε όρισμα ξεχωριστά με την εντολή `sys.argv[1]`. Το `[1]` αντιπροσωπεύει το δεύτερο όρισμα που είναι το `$datapredict` ενώ το όρισμα `[0]` είναι το `python script` (`predict.py`). Τα δεδομένα προς πρόβλεψη (`preddata`) είναι ένα ενιαίο κορδόνι όπου ο κάθε αριθμός χωρίζετε με κόμμα. Για αυτό για να κάνουμε το κάθε αριθμό ξεχωριστό κορδόνι χρησιμοποιούμε τη συνάρτηση `split(',')` με παράμετρο το κόμμα (,) για να μας χωρίσει τους αριθμούς με βάση το κόμμα και θα τους αποθηκεύσουμε στη λίστα `str1`. Στη συνέχεια πρέπει να μετατρέψουμε το `string` σε αριθμούς κινητής υποδιαστολής χρησιμοποιώντας τη συνάρτηση `float()` σε συνδυασμό με το βρόγχο `for`. Τέλος το αποτέλεσμα το αποθηκεύουμε σε έναν πίνακα 8 θέσεων και εκτελείτε ο αλγόριθμος που επέλεξε ο Ιατρός.

## Ιατρικός φάκελος με δυνατότητα πρόβλεψης διαβήτη

```
ver=pd.read_csv("data.csv")
val=[0]*8
#train, test, y_train, y_te

preddata=sys.argv[1]
algorithm=sys.argv[2]
print preddata
str1=preddata.split(',')
print str1[2]
j=0
for i in str1:
    val[j]=float(i)
    j+=1
```

Εικόνα 3.20 Python κώδικας μετατροπείς δεδομένων.

```
if(algorithm == "GNB"):
```

Εικόνα 3.21 Συνθήκη για την εκτέλεση αλγορίθμου..

Τέλος μετά την εκτέλεση του αλγορίθμου η συνάρτηση `shell_exec` επιστρέφει το αποτέλεσμα (τον αλγόριθμο, τη κλάση και τη πιθανότητα) στη μεταβλητή `$flag` και γίνεται μετάβαση στη σελίδα του Ιατρού με τη συνάρτηση `header("Location:doctor_page.php");`. Εκεί μέσω `SESSION` περνάμε το αποτέλεσμα της μεταβλητής `$flag` στη σελίδα του Ιατρού και το αποθηκεύουμε στη μεταβλητή `$string` και με τη συνάρτηση `explode` της PHP το χωρίζουμε σε μικρότερα `string` με βάση το κόμμα και το εμφανίζουμε.

```
if(isset($_SESSION['predict_result'])){
    $string=$_SESSION['predict_result'];
    $predresult=explode(",",$string);

    $string=["Accuracy score:", "Class"];

    for($i=0;$i<2;$i++){
        echo "<tr>
            <td>". $string[$i]. " ". $predresult[$i]. "</td>
            </tr>";
    }
}
```

Εικόνα 3.22 Κώδικας για την εμφάνιση της πρόβλεψης.

## Ιατρικός φάκελος με δυνατότητα πρόβλεψης διαβήτη

### 3.3 Σελίδα ασθενή

#### 3.3.1 Φόρμα εισόδου

Πριν μπει ο ασθενής στη σελίδα ασθενή πρέπει να περάσει πρώτα από τη φόρμα εισόδου. Εκεί πρέπει να βάλει τα στοιχεία του όπως όνομα και κωδικός και αν είναι σωστά θα μεταβεί στη φόρμα ασθενή. Ο έλεγχος των στοιχείων γίνεται προσπελνοντας της εγγραφές του πίνακα patients και συγκρίνοντας τες με τα δεδομένα που έβαλε ο ασθενής.

English  Ελληνικά



The image shows a web form titled "Φόρμα" (Form). It has three input fields: "ΟΝΟΜΑ" (Name), "κωδικός" (Code), and a "Submit" button.

Εικόνα 3.23 Φόρμα εισόδου ασθενή

```
$sql="SELECT pid, name, last_name, pass FROM patients";
$result=$conn->query($sql);

while($row=$result->fetch_assoc()){
    if($row[name] == $name && $row[pass] == $pass){
        $_SESSION['afm'] = $row[pid];
        $_SESSION['pname'] = $row[name];
        $_SESSION['plname'] = $row[last_name];
        header('Location:patient_page.php');
    }
}
```

Εικόνα 3.24 Κώδικας για τον έλεγχο δεδομένων

Αν τα στοιχεία που έβαλε ο ασθενής είναι σωστά τότε τα δεδομένα του θα περάσουν μέσω SESSION στην επόμενη φόρμα στην οποία και θα μεταβεί ο ασθενής.

#### 3.3.2 Φόρμα ασθενή

Μόλις ο ασθενής μεταβεί στη δεύτερη φόρμα θα αντικρίσει ότι το μενού απαρτίζεται από τρεις επιλογές PANTEBOY, ΤΡΕΧΟΝΤΑ PANTEBOY και ΙΣΤΟΡΙΚΟ. Επίσης

**Βασίλειος πεσκελίδης**

## Ιατρικός φάκελος με δυνατότητα πρόβλεψης διαβήτη

τα δεδομένα του ασθενή εμφανίζονται σε πίνακα φέρνοντας το ποντίκι στην επάνω δεξιά εικόνα.



**Εικόνα 3.25** Φόρμα ασθενή

Πατώντας την επιλογή PANTEBOY εμφανίζετε μια φόρμα στην οποία ο ασθενής επιλέγει την ειδικότητα και το όνομα του Ιατρού καθώς και την ημερομηνία που θέλει να κλείσει ο ασθενής το ραντεβού. Τέλος αφού πατήσει υποβολή η εφαρμογή θα εμφανίσει τις διαθέσιμες ώρες για την εκάστοτε ημέρα

Η φόρμα του PANTEBOY αποτελείται από HTML πίνακα και ετικέτα <form>. Στο πρώτο πεδίο ΕΙΔΙΚΟΤΗΤΑ ο ασθενής επιλέγει την ειδικότητα του Ιατρού μέσα από μια ετικέτα <select> με δεδομένα <option>. Η ειδικότητα του Ιατρού παρέχετε από τον πίνακα doctor επιλέγοντας το πεδίο specialty. Επειδή ο πίνακας doctor περιέχει πολλούς Ιατρούς με ίδιες ειδικότητες χρησιμοποιούμε την δήλωση DISTINCT για να μας επιστρέψει μοναδικές εγγραφές. Αφού ο ασθενής επιλέξει την ειδικότητα του Ιατρού και πατήσει υποβολή τότε στην από κάτω ετικέτα <select> θα εμφανιστούν τα ονόματα όλων των Ιατρών που ασχολούνται με την αντίστοιχη ειδικότητα. Στη συνέχεια αφού ο ασθενής επιλέξει το όνομα του Ιατρού, μήνα και ημέρα για το ραντεβού πατάει ξανά υποβολή και η εφαρμογή του εμφανίζει τις διαθέσιμες ώρες για την ημέρα που επέλεξε.

**Εικόνα 3.26** Φόρμα ραντεβού.

Για να εμφανίσουμε τις διαθέσιμες ώρες χρησιμοποιούμε τους πίνακες doctor\_schedule και appointment. Ο πίνακας doctor\_schedule περιέχει τις **Βασίλειος πεσκελίδης**

## Ιατρικός φάκελος με δυνατότητα πρόβλεψης διαβήτη

διαθέσιμες ημέρες και ώρες και ημέρες που μπορεί να δεχτή ασθενής ο Ιατρός. Η ημερομηνία στη φόρμα αποτελείται από τους μήνες ολογράφος και τις ημέρες σε δεκαδική μορφή (10/Οκτωβρίου) γι αυτό για να πάρουμε την ημέρα ολογράφος (Δευτερα) που επέλεξε ο ασθενής χρησιμοποιούμε τη συνάρτηση date() της PHP με παράμετρο μήνα, ημέρα και έτος. Στη συνέχεια αφού έχουμε την ημέρα ολογράφος κάνουμε ένα ερώτημα προς τη βάση επιλέγοντας από το πίνακα doctor\_schedule το πεδίο τις ημέρας που μας επέστρεψε η συνάρτηση date() και το πεδίο id που είναι ο αριθμός του Ιατρού Έτσι από το ερώτημα έχουμε τις διαθέσιμες ώρες του Ιατρού για την αντίστοιχη ημέρα Ύστερα κάνουμε ερώτημα προς τον πίνακα appointment και επιλέγουμε τις ώρες που αντιστοιχούν στο μήνα, ημέρα και το αριθμό του Ιατρού Τέλος συγκρίνουμε με συνθήκη if και δυο βρόγχους τις υπάρχουσες ώρες από τον πίνακα appointment με αυτές του doctor\_schedule και αν η ώρες διαφέρουν τότε εμφανίζονται ως διαθέσιμες.

```
<?php
$y=date("Y");
$day=date("l", mktime(0, 0, 0, $month, $x, $y));

$mnth=$monthapp;
$sqlq3="SELECT ". $day ."
        FROM doctor_schedule
        WHERE doctor_schedule.id='".$ $names ."'";
$result3=$conn->query($sqlq3);

while($row=$result3->fetch_assoc()){

    $sqlq4="SELECT timep FROM appointment
            WHERE appointment.d_id='".$ $names ."' AND appointment.month='".$ $month ."' AND appointment.day='".$ $x ."'";
    $result4=$conn->query($sqlq4);

    while($row2=$result4->fetch_assoc()){
        if($row[$day] == $row2[timep]){
            $flag=1;
            break;
        }

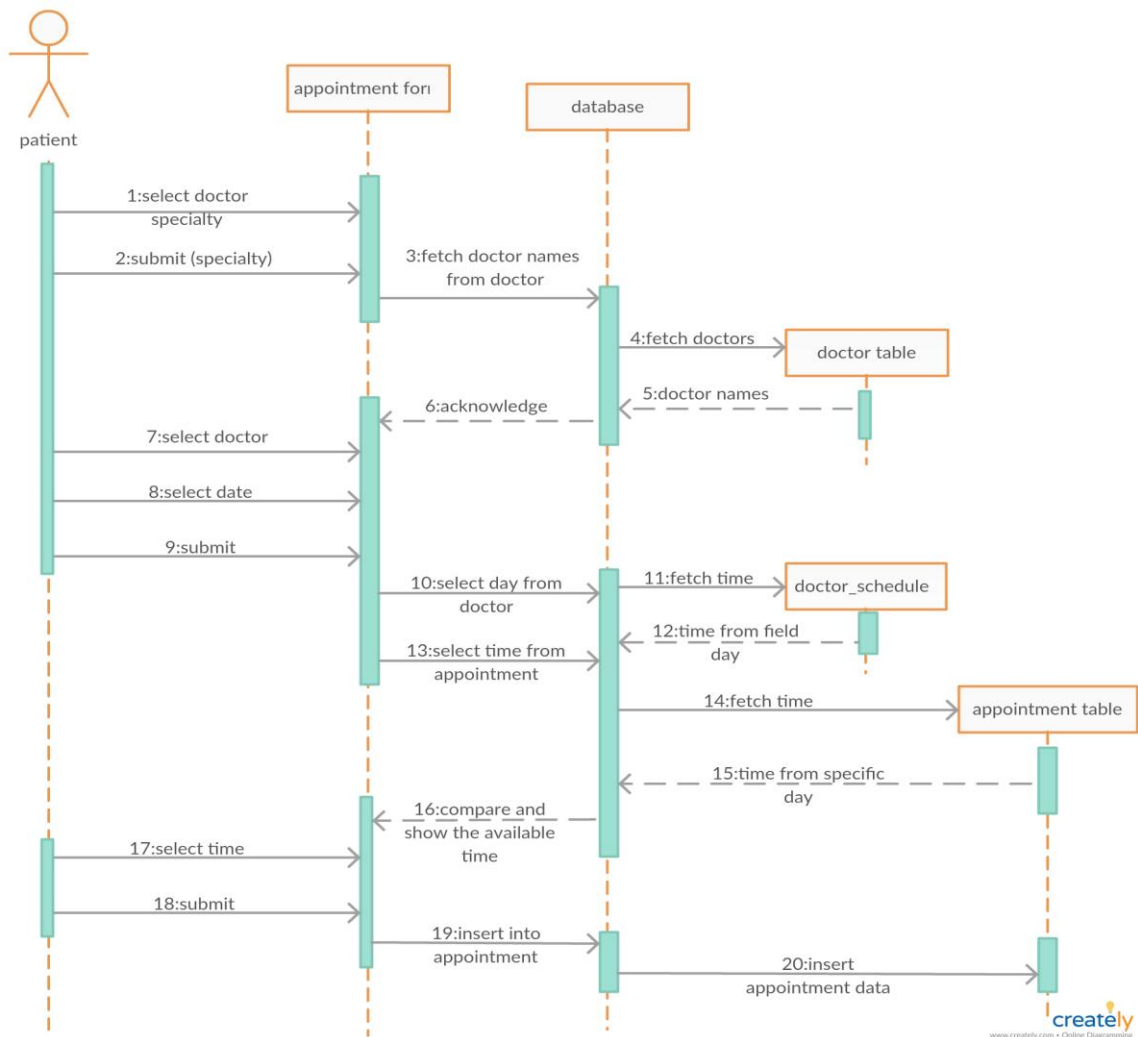
        else
            $flag=0;
    }

    if($flag == 0)
        echo "<option value='".$ $row[$day]."'>".$row[$day]."</option>";
}
```

**Εικόνα 3.27** Κώδικας για τις διαθέσιμες ώρες.

Όταν ο ασθενής επιλέξει την ώρα και πατήσει υποβολή τα δεδομένα που επέλεξε στη φόρμα θα αποθηκευτούν στον πίνακα appointment.

## Ιατρικός φάκελος με δυνατότητα πρόβλεψης διαβήτη



**Εικόνα 3.28**UML διάγραμμα ακολουθίας ραντεβού.

Τα κουτιά με τις διακεκομμένες γραμμές λέγονται lifeline και είναι κάποιος ρόλος ή στιγμιότυπο αντικειμένου και οι έντονες πράσινες γραμμές σημαίνουν ότι γίνεται κάποια ενέργεια ή ενεργοποίηση. Επίσης για να δείξουμε την επικοινωνία χρησιμοποιούμε γραμμές με βέλη και οι διακεκομμένες γραμμές σημαίνουν ανταπόκριση από το σύστημα. Τέλος το διάγραμμα αποτελείται από πέντε lifeline τρεις πίνακες μια βάση δεδομένων και μια φόρμα ραντεβού.



## Ιατρικός φάκελος με δυνατότητα πρόβλεψης διαβήτη

Στην επιλογή ΤΡΕΧΟΝΤΑ PANTEBOY ο ασθενής μπορεί να δει τα μελλοντικά του ραντεβού που έχει κλίση, επίσης τα ραντεβού είναι ταξινομημένα κατά αύξουσα ημερομηνία. Μόλις μπει ο ασθενής στη σελίδα μετά τη φόρμα εισόδου παίρνουμε το AMKA του μέσω SESSION. Ύστερα κάνουμε ερώτημα προς προς τη βάση επιλέγοντας την ημερομηνία και την ώρα από το πίνακα appointment ανάλογα με το AMKA του ασθενή καθώς και το όνομα, επίθετο και την ειδικότητα του Ιατρού από το πίνακα doctor. Ο πίνακας appointment έχει δυο ξένα κλειδιά το πεδίο pid που είναι το πρωτεύον κλειδί στο πίνακα patient και είναι το AMKA του ασθενή και τέλος το πεδίο d\_id και είναι το πρωτεύον κλειδί στο πίνακα doctor και χαρακτηρίζει μοναδικά τον κάθε Ιατρό. Αφού οι πίνακες συσχετίζονται μεταξύ τους επιλέγουμε τα πεδία που θέλουμε και εφαρμόζουμε JOIN.

```
$sql3="SELECT
  doctor.first_name,
  doctor.sur_name,
  doctor.specialty,
  appointment.pid,
  appointment.date,
  appointment.timep
FROM appointment
INNER JOIN doctor
ON appointment.d_id = doctor.D_id
WHERE appointment.date >= CURDATE() AND appointment.pid = '$_SESSION['afm'].'";
ORDER BY appointment.`date`";

$result3=$conn->query($sql3);
while($row3=$result3->fetch_assoc()){
  echo "<tr>
    <td>". $row3['first_name']. "</td>
    <td>". $row3['sur_name']. "</td>
    <td>". $row3['specialty']. "</td>
    <td>". $row3['date']. "</td>
    <td>". $row3['timep']. "</td>
  <tr>";
}
```

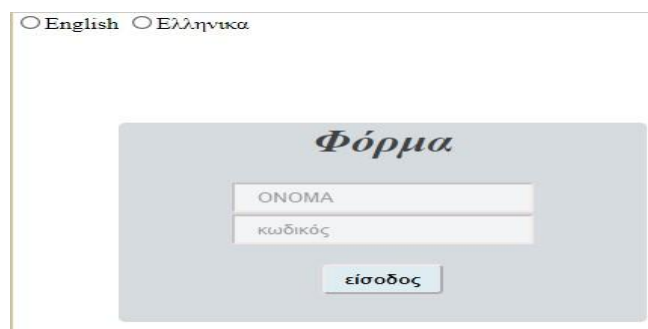
**Εικόνα 3.29**Κώδικας για τα τρέχοντα ραντεβού.

Τέλος πατώντας την επιλογή ΙΣΤΟΡΙΚΟ εμφανίζετε το ιστορικό του ασθενή σε μορφή HTML πίνακα Τα δεδομένα του HTML πίνακα είναι το αποτέλεσμα τριών πινάκων (doctor, examination,patient) και δυο JOIN. Επίσης χρησιμοποιούμε WHERE για να πάρουμε τα αποτελέσματα μόνο του αντίστοιχου ασθενή με βάση το AMKA.

## 3.4 Φόρμα γραμματείας

### 3.4.1 Φόρμα εισόδου

Προκειμένου η γραμματεία να μπει στη κύρια φόρμα διαχείρισης πρέπει πρώτα να πέραση από τη φόρμα εισόδου. Εκεί η γραμματεία θα βαλή τα στοιχεία στη φόρμα και αν τα δεδομένα που έβαλε είναι σωστά τότε θα αποθηκευθούν στον πίνακα SESSION και μετά θα μεταβεί στην επόμενη σελίδα



**Εικόνα 3.30** Φόρμα εισόδου γραμματείας.

Για τον έλεγχο των δεδομένων θα προσπελάσουμε τον πίνακα secretary και επιλέγουμε τα πεδία όνομα, επίθετο, κωδικό και το id. Τέλος συγκρίνουμε τα δεδομένα του πίνακα με αυτά που έβαλε η γραμματεία.

```
$sql = "SELECT id, first_name, last_name, password FROM secretary";
$result = $conn->query($sql);

if ($result->num_rows > 0) {
    // output data of each row
    while($row = $result->fetch_assoc()) {
        $f_name=$row["first_name"];
        $l_name=$row["last_name"];
        $pas=$row["password"];
        $id=$row["id"];

        if(isset($f_name) == isset($name) && $pas == $pass){
            $_SESSION['lang'] = "$sub";
            $_SESSION['name'] = "$f_name";
            $_SESSION['lname'] = "$l_name";
            #$_SESSION['id'] = "$id";
            header('Location:secretary.php');
        }
    }
}
```

**Εικόνα 3.31** Κώδικας ελέγχου δεδομένων.

# Ιατρικός φάκελος με δυνατότητα πρόβλεψης διαβήτη

## 3.4.2 Φόρμα γραμματείας

Το μενού τις γραμματείας αποτελείτε από τις τρεις επιλογές ΔΙΑΓΡΑΦΗ ΔΕΔΟΜΕΝΩΝ, ΠΛΗΡΟΦΟΡΙΕΣ ΑΣΘΕΝΗ και ΕΓΓΡΑΦΗ.



Εικόνα 3.32 Σελίδα γραμματείας.

Στην επιλογή ΔΙΑΓΡΑΦΗ ΔΕΔΟΜΕΝΩΝ η γραμματεία μπορεί να διαγράψει μια ή πολλαπλές εγγραφές από το ιστορικό του ασθενή Πατώντας αυτή την επιλογή εμφανίζετε ένα μενού εκεί η γραμματεία συμπληρώνει ένα πεδίο με το ΑΜΚΑ του ασθενή και πατάει υποβολή. Ύστερα εμφανίζετε ένας πίνακας με το ιστορικό του ασθενή και για να διαγράψουμε κάποια εγγραφή πρέπει πρώτα να επιλέξουμε τις εγγραφές που θέλουμε πατώντας με το ποντίκι το checkbox στο πεδίο CHECK και μετά να πατήσουμε το κουμπί ΔΙΑΓΡΑΦΗ.

CHECK	ID	Examination	Time	Diagnosis	Drug	Dossage	Date	Side_effects	Doctor_name	Doctor_last_name	SPECIALTY
<input type="checkbox"/>	1	KARDIA	20:11:33	XOLISTERINI	ASPIRINI	1/DAY	2018-10-03	---	VASILIS	PESKELIDIS	cardiologist

Εικόνα 3.33 Μενού διαγραφής δεδομένων.

```
doctor.first_name,  
doctor.sur_name,  
doctor.specialty,  
examination.exam,  
examination.results,  
examination.notes,  
examination.drug,  
examination.dossage,  
examination.side_effects,  
examination.date,  
examination.time,  
examination.id,  
patients.pid  
FROM  
doctor  
INNER JOIN examination ON doctor.D_id = examination.D_id  
INNER JOIN patients ON examination.pid = patients.pid  
WHERE patients.pid = '$pi.';";  
//$sqlq="SELECT pname, piname, med, dos, dayyy, ti, dname, dlname FROM history";  
$result = $conn->query($sqlq2);  
while($row = $result->fetch_assoc()) {  
    echo"<tr>  
    <td><input type='checkbox' name='id[]' value='." . $row["id"] . "></td>  
    <td align='center'>". $row["pid"] . "</td>  
    <td align='center'>". $row["exam"] . "</td>  
    <td align='center'>". $row["notes"] . "</td>  
    <td align='center'>". $row["drug"] . "</td>  
    <td align='center'>". $row["side_effects"] . "</td>  
    <td align='center'>". $row["dossage"] . "</td>  
    <td align='center'>". $row["date"] . "</td>  
    <td align='center'>". $row["time"] . "</td>  
    <td align='center'>". $row["first_name"] . "</td>  
    <td align='center'>". $row["sur_name"] . "</td>  
    <td align='center'>". $row["specialty"] . "</td>  
    </tr>";  
}
```

Εικόνα 3.34 Κώδικας διαγραφής δεδομένων.

## Ιατρικός φάκελος με δυνατότητα πρόβλεψης διαβήτη

Στην επόμενη επιλογή ΠΛΗΡΟΦΟΡΙΕΣ ΑΣΘΕΝΗ η γραμματεία μπορεί να δει τα δεδομένα του ασθενή όπως τις προσωπικές πληροφορίες αρκεί να βάλει το ΑΜΚΑ του ασθενή στο πεδίο αναζήτησης και να πατήσει υποβολή. Τα δεδομένα του ασθενή τα παίρνουμε από το πίνακα patients με βάση το ΑΜΚΑ και επιλέγοντας τα πεδία που θέλουμε.

ΑΜΚΑ	SEARCH				
Patient_name	Patient_last_name	birthday	phone_number	address	password
NIKOS	SOULIS	15/3/1985	234234234	andromaxis 25 glifada	1232

**Εικόνα 3.35** Μενού αναζήτησης πληροφοριών ασθενή.

Τέλος στην επιλογή ΕΓΓΡΑΦΗ η γραμματεία εισάγει τα δεδομένα του ασθενή και τον εγγράφει στο σύστημα πατώντας υποβολή. Τα δεδομένα που εισάγει η γραμματεία είναι το ονοματεπώνυμο, ημερομηνία γέννησης, τηλέφωνο και διεύθυνση κατοικίας. Επίσης στο πεδίο κωδικός η γραμματεία ζητάει από τον ασθενή να βάλει έναν κωδικό με τον οποίο ο ασθενής θα μπαίνει στη σελίδα ασθενή.

ΥΠΟΒΟΛΗ	
ΑΜΚΑ	<input type="text"/>
ΟΝΟΜΑ_ΑΣΘΕΝΗ	<input type="text"/>
ΕΠΙΘΕΤΟ_ΑΣΘΕΝΗ	<input type="text"/>
ΚΩΔΙΚΟΣ	<input type="text"/>
ΗΜΕΡΟΜΗΝΙΑ ΓΕΝΝΗΣΗΣ	<input type="text"/>
ΤΗΛΕΦΩΝΩ	<input type="text"/>
ΔΙΕΥΘΥΝΣΗ	<input type="text"/>

**Εικόνα 3.36** Μενού εγγραφής ασθενή.

## ΚΕΦΑΛΑΙΟ 4

### Πρόβλεψη διαβήτη

#### Εισαγωγή

Στο κεφάλαιο αυτό θα αναφερθούμε στη τεχνητή νοημοσύνη συγκεκριμένα στη μηχανική μάθηση. Επίσης θα παρουσιάσουμε τη πρόβλεψη διαβήτη μαζί με τους τρεις αλγόριθμους (Naive Bayes, support vector machine, Random forest) καθώς και τους μαθηματικούς τύπους. Τέλος θα παρουσιάσουμε ορισμένους αλγορίθμους μηχανικής μάθησης και θα αναφερθούμε στη βαθιά μάθηση και στην ενισχυμένη μάθηση και τις εφαρμογές τους.

#### 4.1 Ιστορική αναδρομή

Το 1950 ο Άγγλος μαθηματικός Alan Turing παρουσίασε στο παπτερ του “Computing Machinery and Intelligence” το Turing test ή το imitation game όπως το ονόμασε και έθεσε το ερώτημα μπορούν οι μηχανές να σκεφτούν. Το τεστ αυτό έχει σαν σκοπό να δοκιμάσει την ικανότητα της μηχανής εάν παρουσιάζει ευφυΐα ίση με αυτή του ανθρώπου. Η δοκιμασία αυτή γίνεται ως εξής, ένα μεγάλο δωμάτιο χωρισμένο σε τρία μικρότερα δωμάτια όπου στο ένα μπαίνει ο άνθρωπος και στο άλλο η μηχανή. Τέλος στο τρίτο δωμάτιο μπαίνει ένας άλλος άνθρωπος ο οποίος κάνει ερωτήσεις προς τον άνθρωπο και τη μηχανή χωρίς να ξέρη σε ποια δωμάτια βρίσκονται. Στο τέλος του τεστ αν δεν διαπιστώθηκε διαφορά μεταξύ ανθρώπου και υπολογιστή τότε η μηχανή περνάει το Turing test.

Το 1951 ο Γερμανός επιστήμονας πληροφορικής Dietrich Gunther Prinz έγραψε το πρώτο πρόγραμμα για σκάκι. Ενώ στις αρχές της δεκαετίας του 60' ο Αμερικάνος Arthur Lee Samuel έγραψε πρόγραμμα για το παιχνίδι ντάμα το οποίο ανέπτυξε ικανότητες ώστε να παίζει με αρχάριους παίκτες.

Το 1964 ο Αμερικάνος επιστήμονας Daniel G. Bobrow έφτιαξε το πρόγραμμα STUDENT το οποίο διαβάζει και λύνει αλγεβρικά προβλήματα. Το πρόγραμμα **Βασίλειος πεσκελίδης**

## Ιατρικός φάκελος με δυνατότητα πρόβλεψης διαβήτη

αυτό θεωρήθηκε η πρώτη επιτυχία στο κλάδο τις φυσικής επεξεργασία γλώσσας. Στο MIT εργαστήριο τεχνίτης νοημοσύνης δημιουργήθηκε το πρόγραμμα φυσικής επεξεργασίας γλώσσας ELIZA το 1966 από το Joseph Weizenbaum. Ο οδηγίες για τον πρώτο αλληλεπίδρασης παρέχονταν από scrip γραμμένα σε MAD-Slip.

Το 1965 ο Ουκρανός μαθηματικός Alexey Ivakhnenko δημοσίευσε το πρώτο λειτουργικό τεχνητό νευρωνικό δίκτυο πολλών επιπέδων. Στο Waseda University τις Ιαπωνίας το 1972 φτιάξανε το WABOT-1 το πρώτο πλήρες ευφυές ρομπότ το οποίο περπατούσε και μπορούσε να πιάνει και να μεταφέρει αντικείμενα και η όραση του επέτρεπε να μετράει την απόσταση μεταξύ αντικειμένων. Επίσης είχε τεχνητά αυτιά και μπορούσε να μιλάει μέσω τεχνητού στόματος. Μαθητές του πανεπιστήμιου Stanford κατασκεύασαν το 1979 ένα καρτ το οποίο μπορούσε να πληγείτε και να αποφεύγει μονό του τα εμπόδια.

Το 1980 κατασκευαστικέ το πρώτο έμπειρο σύστημα XCON και λειτουργούσε στην Digital Equipment Corporation. Ήταν μεγάλη επιτυχία για την εταιρία γιατί μέχρι 1989 η εταιρία εξοικονομούσε 40 εκατομμύρια δολάρια ετησίως. Στη δεκαετία του 90' γίνεται ευρέως γνωστή η ιδέα του ευφυή πράκτορα ο οποίος χρησιμοποιούνταν για εξόρυξη υπηρεσίες ειδήσεων και online αγορές. Επίσης την ίδια εποχή η τεχνητή νοημοσύνη εφαρμόζετε σε συστήματα logistics, εξόρυξη δεδομένων και για ιατρική διάγνωση. Οι επιτυχίες αυτές οφείλονται χάρις την αυξημένη υπολογιστική ισχύ διότι για την εκτέλεση τέτοιων αλγορίθμων υπήρχαν απαίτησης ισχύος.

Τον Φεβρουάριο του 2010 η Apple παρουσίασε στην αγορά την Siri. Η είναι ένας εικονικός βοηθός και υπακούει στα φωνητικά ερωτήματα και τα επεξεργάζεται μέσω τις φυσικής επεξεργασίας γλώσσας. Το 2011 το Apache Software Foundation κατασκεύασε το apache hadoop το οποίο παρέχει υπηρεσίες επεξεργασίας τεράστιου όγκου δεδομένων (big data) τα οποία αποθηκεύονται στο Hadoop Distributed File System και επεξεργάζονται με τις τεχνικές MapReduce. Το 2015 η Google κατασκεύασε το Tensorflow μια βιβλιοθήκη ανοικτού κώδικα για τον προγραμματισμό ροής δεδομένων και χρησιμοποιείτε για τον προγραμματισμό στη μηχανική μάθηση και στη βάθια μάθηση

## Ιατρικός φάκελος με δυνατότητα πρόβλεψης διαβήτη

### 4.2 Μηχανική μάθηση

Η μηχανική μάθηση είναι κλάδος της τεχνητής νοημοσύνης και χρησιμοποιεί τεχνικές στατιστικής για να δώσει τη δυνατότητα στους υπολογιστές να μαθαίνουν από δεδομένα και να κάνουν προβλέψεις. Στη μηχανική μάθηση υπάρχουν τρις τύποι αλγορίθμων supervised learning, unsupervised learning και reinforcement learning.

- Supervised learning

Ο αλγόριθμος προσπαθεί να προβλέψει μια μεταβλητή στόχου (label) από ένα δεδομένο σύνολο μεταβλητών (features). Δηλαδή, πριν φτιάξουμε το μοντέλο πρόβλεψης πρέπει να χωρίσουμε τα δεδομένα σε εκπαίδευση και τεστάρισμα. Ύστερα φτιάχνουμε το μοντέλο και του περνάμε τα δεδομένα εκπαίδευσης ο αλγόριθμος μαθαίνει από αυτά τα δεδομένα και όταν του δώσουμε τα δεδομένα τεστάρισμα θα εκτιμήσουμε την απόδοση του μοντέλου.

Όταν η προβλεπόμενη μεταβλητή είναι κατηγορία (ναι, όχι, 1, 0) τότε το πρόβλημα είναι ταξινόμησης (classification). Διαφορετικά εάν έχει αριθμητικές τιμές (34.67) τότε είναι regression. Παραδείγματα supervised learning αλγορίθμων: Regression, Decision Tree, Random Forest, KNN, Logistic Regression.

Παράδειγμα: Τα δεδομένα χωρίζονται σε train test και predict. Ο αλγόριθμος εκπαιδεύεται με τα δεδομένα train ενώ με τα δεδομένα test θα πρέπει να προβλέψει τα predict με βάση τα δεδομένα train.

	34	44	0	
	34	64	1	
	55	56	1	
TRAIN	89	54	1	
	88	55	0	
TEST				PREDICT

Πίνακας 4.1 Χωρισμός δεδομένων.

## Ιατρικός φάκελος με δυνατότητα πρόβλεψης διαβήτη

- Unsupervisedlearning

Σε αυτήν τη περίπτωση δεν έχουμε μεταβλητή στόχου (label) παρά μόνο τις μεταβλητές. Προκειμένου να βγάλουμε χρήσιμη πληροφορία από τα δεδομένα χρησιμοποιούμε τη μέθοδο clustering. Η μέθοδος αυτή προσπαθεί να βρει μια συσχέτιση μεταξύ των δεδομένων για να τα ομαδοποιήσει, δηλαδή ο αλγόριθμος προσπαθεί να εξάγει πληροφορία από άγνωστα δεδομένα χωρίς προβλεπόμενη μεταβλητή. Παραδείγματα Unsupervisedlearning αλγορίθμων: Apriorialgorithm, K-means.

- Reinforcementlearning

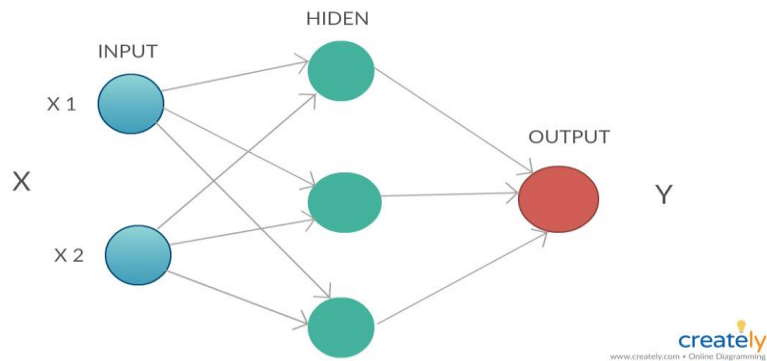
Χρησιμοποιώντας αυτόν τον αλγόριθμο η μηχανή εκπαιδεύεται να παίρνει συγκεκριμένες αποφάσεις. Η μηχανή εκτίθεται σε περιβάλλον όπου εκπαιδεύεται συνεχώς χρησιμοποιώντας δοκιμές, σφάλματα και μαθαίνει από την παρελθοντική εμπειρία η κατάσταση και προσπαθεί να καταγράψει τη καλύτερη γνώση. Παράδειγμα Reinforcementlearning αλγορίθμων: MarkovDecisionProcess.

Χάρης τη μηχανική μάθηση δημιουργήθηκε ο κλάδος της εξόρυξης δεδομένων. Η εξόρυξη δεδομένων (datamining) ασχολείται με την ανακάλυψη γνώσης από βάσεις τεράστιου όγκου δεδομένων μέσω αλγορίθμων. Στόχος της είναι η πληροφορία που θα εξαχθεί από τις βάσεις να είναι δομημένη και κατανοητή προς τον άνθρωπο για να τον βοηθήσει να πάρει τις κατάλληλες αποφάσεις. Υπάρχουν 5 στάδια για την ανακάλυψη γνώσης από δεδομένα: συλλογή, προεπεξεργασία, μετασχηματισμός, εξόρυξη δεδομένων και αξιολόγηση. Μαζί με το datamining εντάσσεται και ο όρος bigdata ο οποίος αναφέρετε σε δεδομένα μεγάλης κλίμακας. Αυτά τα δεδομένα συλλέγονται από διάφορες πηγές και είναι διαφορετικού τύπου. Οι τρεις τύποι των δεδομένων στα bigdata είναι δομημένα, ημιοδομημένα και μη δομημένα. Το bigdataanalytics χρησιμοποιείται ευρέως από εταιρίες για να κάνουν καλύτερη την αγορά τους μέσα από την ανάλυση των δεδομένων μεγάλης κλίμακας. Ορισμένα εργαλεία για τη διαχείριση και επεξεργασία bigdata: Hadoop, Pig, Hive, Cassandra, Spark, Kafka.



## Ιατρικός φάκελος με δυνατότητα πρόβλεψης διαβήτη

Ένας άλλος κλάδος της μηχανικής μάθησης είναι η βαθιά μάθηση (deep learning) και προσπαθεί να μοντελοποιήσει τον τρόπο που ο ανθρώπινος εγκέφαλος επεξεργάζεται τα δεδομένα. Μια αρχιτεκτονική βαθιάς μάθησης είναι τα τεχνητά νευρωνικά δίκτυα τα οποία προσομοιώνουν την λειτουργία ενός νευρώνα.



**Εικόνα 4.1** Τεχνητό νευρικό δίκτυο.

Το δίκτυο απαρτίζεται από μονάδες εισόδου βάρους πόλωσης  $X_1$ ,  $X_2$  και συνδέονται με κατευθυνόμενους συνδέσμους με κρηφο δίκτυο και από εκεί στην έξοδο  $Y$ . Σε κάθε σύνδεσμο ή κόμβο υπάρχει μια συνάρτηση ενεργοποίησης η οποία ενεργοποιείται ανάλογα με την είσοδο που δέχεται και η τιμή ενεργοποίησης είναι 1 όταν είναι σωστή η είσοδος και 0 όταν είναι λάθος.

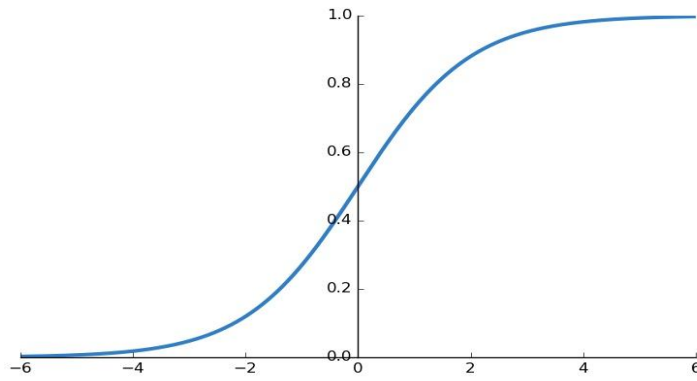
Μια από τις συναρτήσεις ενεργοποίησης είναι η sigmoid και παίρνει σαν παράμετρο το άθροισμα γινομένου εισόδου επί το βάρος

$$\sum_j^n W_{j,ixj}$$

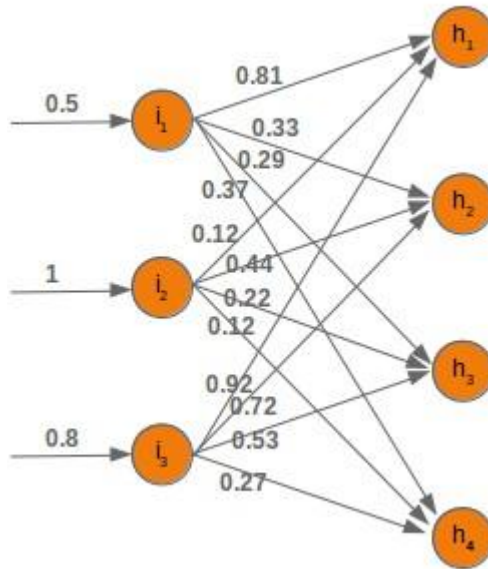
**Εικόνα 4.2** Άθροισμα εισόδων

Όπου το  $x_j$  είναι η είσοδος ( $x_1$ ,  $x_2$ ) και  $W_{i,j}$  το βάρος πόλωσης. Το άθροισμα αυτό είναι είσοδος στη συνάρτηση ενεργοποίησης (sigmoid) και περιγράφεται με μαθηματικό τύπο  $1/1+e^{-x}$ . Η συνάρτηση αυτή μετατρέπει οποιαδήποτε τιμή μεγαλύτερη του 0.5 σε 1 και διαφορετικά σε 0. Στην εικόνα 4.3 ο οριζόντιος άξονας είναι η είσοδος και ο κάθετος η έξοδος.

## Ιατρικός φάκελος με δυνατότητα πρόβλεψης διαβήτη



Εικόνα 4.3 Συνάρτησης sigmoid.



Εικόνα 4.4 Είσοδοι με βάρη σε κόμβους.

Παράδειγμα για τον κόμβο  $h_1$ :  $z=0.81*0.5+0.12*1+0.92*0.8$ ,  $\text{sigmoid}(z)$ .

```
import numpy as np
z=0.81*0.5+0.12*1+0.92*0.8

y=(1 / (1 + np.exp(-z)))
print y
if(y > 0.5):
    print 1
else:
    print 0
```

Εικόνα 4.5 Sigmoid με python.

## Ιατρικός φάκελος με δυνατότητα πρόβλεψης διαβήτη

### 4.3 Πρόβλεψη διαβήτη

Όταν ο Ιατρός εισάγει τα δεδομένα προς πρόβλεψη και επιλέξει τον αλγόριθμο που θέλει τότε εκτελείτε το `pythonscript`. Μέσα σε αυτό το αρχείο φορτώνονται τα δεδομένα από ένα εξωτερικό αρχείο CSV (CommaSeparatedValues) με τις μετρήσεις από διαβήτη για την εκπαίδευση και το τεστάρισμα του εκάστοτε αλγόριθμου.

Τα δεδομένα προς εκπαίδευση του αλγόριθμου τα φορτώνουμε με τη συνάρτηση `read_csv()` και με την βιβλιοθήκη `pandas`. Η συνάρτηση δέχεται ως παράμετρο το όνομα του αρχείου.

```
import pandas as pd
data=pd.read_csv("data.csv")
```

**Εικόνα 4.6** Φόρτωμα δεδομένων από εξωτερικό αρχείο

Αφού φορτώσουμε τα δεδομένα πρέπει να τα χωρίσουμε σε εξαρτημένες και ανεξάρτητες μεταβλητές. Δηλαδή χωρίζουμε τις προβλεπόμενες μεταβλητές από τα υπόλοιπα δεδομένα.

```
y=data.feature
x=data.drop('feature', axis=1)
```

**Εικόνα 4.7** Όπου  $x$  εξαρτημένες μεταβλητές και  $y$  ανεξάρτητες.

Με την συνάρτηση `drop()` κόβουμε την στυλή `feature` από τα δεδομένα η οποία είναι η στυλή που θέλουμε να προβλέψουμε. Η παράμετρος `axis=1` δηλώνει ότι αναφερόμαστε σε στυλή ενώ το `axis=0` σε γραμμή

Στη συνέχεια χωρίζουμε τα δεδομένα σε εκπαίδευση και τεστάρισμα με τη συνάρτηση `train_test_split` από τη βιβλιοθήκη `sklearn`. Οι παράμετροι της συνάρτησης είναι οι εξαρτημένες και η ανεξάρτητη μεταβλητή, καθώς σε τη ποσοστό θέλουμε να χωρίσουμε τα δεδομένα προς εκπαίδευση και τεστάρισμα.

```
x_train, x_test, y_train, y_test = train_test_split(x,y, test_size=0.2, random_state=0)
```

**Εικόνα 4.8** Δεδομένα προς εκπαίδευση και τεστάρισμα.

## Ιατρικός φάκελος με δυνατότητα πρόβλεψης διαβήτη

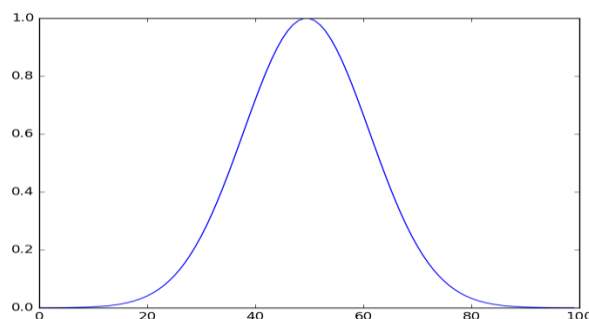
### 4.3.1 Gaussian Naive Bayes

Ύστερα εκτελείται ο πρώτος αλγόριθμος από τους τρεις ο Gaussian Naive Bayes καλώντας τη συνάρτηση GaussianNB() και εκχωρώντας το μοντέλο στη μεταβλητή model. Στη συνέχεια μέσω της συνάρτησης fit() εκχωρούμε στον αλγόριθμο τα δεδομένα προς μάθηση για να το εκπαιδεύσουμε και στο επόμενο βήμα ελέγχουμε την αποδοτικότητα του αλγόριθμου δίνοντας του τα δεδομένα προς πρόβλεψη με την συνάρτηση predict(). Επίσης η μεταβλητή val περιέχει τα δεδομένα που εισήγαγε ο Ιατρός προς πρόβλεψη. Τέλος για να χρησιμοποιήσουμε τη συνάρτηση GaussianNB() πρέπει να συμπεριλάβουμε από την βιβλιοθήκη Sklearn το GaussianNB.

```
from sklearn.naive_bayes import GaussianNB
model= GaussianNB()
model.fit(train,y_train)
predictions=model.predict(test)
print "%.3f" % accuracy_score(y_test, predictions)
predictions =model.predict([val])
print ", ",predictions
```

**Εικόνα 4.9** Μοντέλο του αλγορίθμου GaussianNaiveBayes.

Στη μηχανική μάθηση ο Naive Bayes είναι αλγόριθμος ταξινόμησης βασισμένος στο θεώρημα Bayes και προβλέπει τις ανεξάρτητες μεταβλητές με βάση τις εξαρτημένες. Πριν γίνει η εκτίμηση εφαρμόζεται η συνάρτηση Gauss στα δεδομένα για να βρούμε την μέση τιμή και την κανονική κατανομή των δεδομένων και να πολλαπλασιάσουμε τις συνάρτησης GAUSS μεταξύ τους που θα προκύψουν από αναπαράσταση των μεταβλητών.



**Εικόνα 4.8** Συνάρτηση Gauss.

## Ιατρικός φάκελος με δυνατότητα πρόβλεψης διαβήτη

Η συνάρτηση Gauss εκφράζει την πιθανότητα πυκνότητας μιας κανονικής κατανομής.

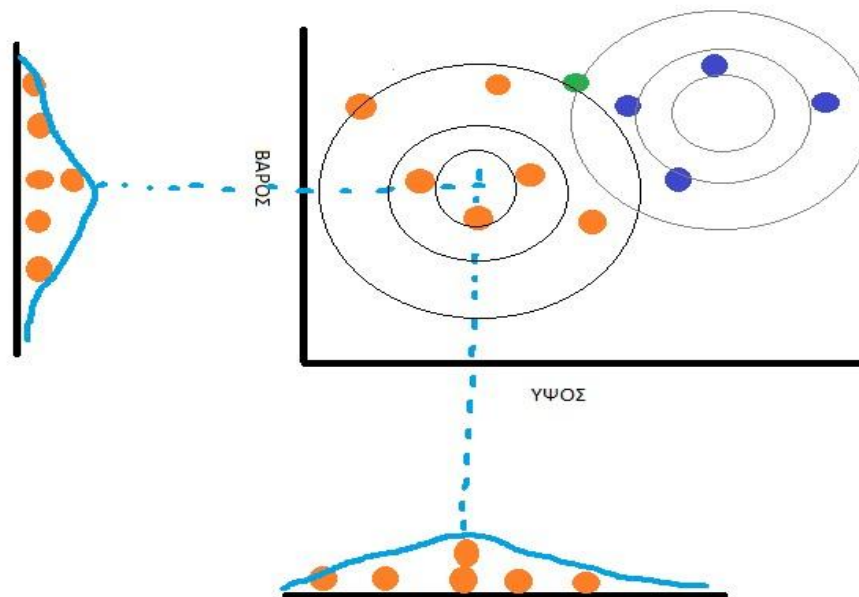
$$g(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2}.$$

**Εικόνα 4.9** Εξίσωση Gauss.

Όπου το  $\sigma$  είναι η κανονική κατανομή και  $\mu$  η μέση τιμή. Για τη μέση τιμή έχουμε  $\mu = \sum z_i / N$  όπου  $N$  είναι ο αριθμός των δειγμάτων και για την κατανομή  $\sigma = \sqrt{\sum (z_i - \mu)^2 / N}$ .

**ΠΑΡΑΔΕΙΓΜΑ:** έστω ότι έχουμε 6 δείγματα από παιδιά (C) και 4 δείγματα με ενήλικες (A) με το βάρος (W) και το ύψος (H) τους και θέλουμε να προβλέψουμε σε τι κλάση ανήκει το νέο δείγμα

Το ποσοστό των ενηλίκων είναι  $P(A) = 4/4+6 = 0.4$  (40%) και των παιδιών  $P(C) = 6/4+6 = 0.6$  (60%).



**Εικόνα 4.10** διάγραμμα Gauss.

Πολλαπλασιάζοντας τα δυο διαγράμματα GAUSS προκύπτουν οι κύκλοι και τα δεδομένα που είναι πιο κοντά στο κέντρο του κύκλου έχουν μεγαλύτερο ποσοστό εκτιμήσεις.

**Βασίλειος πεσκελίδης**

## Ιατρικός φάκελος με δυνατότητα πρόβλεψης διαβήτη

Έτσι για να εκτιμήσουμε σε τη κλάση ανήκει το νέο δείγμα (πράσινη κουκκίδα) πρέπει να πάρουμε της εκτιμήσεις για την κάθε κλάση και της εξαρτημένες μεταβλητές (βάρος, ύψος).

$$P(H|C) = \frac{1}{\sqrt{2\pi}\sigma_{H,C}} e^{-\frac{(z-\mu_{H,C})^2}{2\sigma_{H,C}^2}}$$
$$P(W|C) = \frac{1}{\sqrt{2\pi}\sigma_{W,C}} e^{-\frac{(z-\mu_{W,C})^2}{2\sigma_{W,C}^2}}$$
$$P(H|A) = \frac{1}{\sqrt{2\pi}\sigma_{H,A}} e^{-\frac{(z-\mu_{H,A})^2}{2\sigma_{H,A}^2}}$$
$$P(W|A) = \frac{1}{\sqrt{2\pi}\sigma_{W,A}} e^{-\frac{(z-\mu_{W,A})^2}{2\sigma_{W,A}^2}}$$

---

**Εικόνα 4.11** Πιθανότητες άπω την εξίσωση Gauss.

Ύστερα συνδυάζουμε τις πιθανότητες που βρήκαμε στον κανόνα Bayes

$P(A|z) = P(z|A)P(A)/P(z)$  η πιθανότητα τις A υπό τη συνθήκη τις B. Άρα έχουμε

$$P(z|A) = P(Wz|A) P(Hz|A)$$

$$P(z|C) = P(Wz|C) P(Hz|C)$$

Τέλος συνδυάζουμε μαζί τις δυο εξίσωσης για να βρούμε την πιθανότητα το

δείγμα να ανήκει στη κλάση ενήλικας:  $P(A|z) = P(z|A) P(A) / P(z|A) P(A) + P(z|C)$

$$P(C)$$

Εκτελώντας τον αλγόριθμο για τα δεδομένα: 7,95,79,33,0,37.7,0.370,27 το αποτέλεσμα είναι η κλάση μηδέν (0) που σημαίνει ότι δεν είναι διαβητικός ο ασθενής. Τέλος η επίδοση του αλγορίθμου περιγράφεται από το Accuracy score..

## Ιατρικός φάκελος με δυνατότητα πρόβλεψης διαβήτη

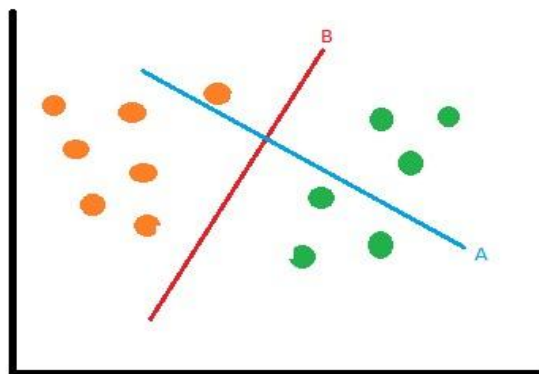
Accuracy score:0.766
Class [0]

**Εικόνα 4.12** Αποτέλεσμα του GaussianNaiveBayes.

### 4.3.2 SVM (SupportVectorMachine)

Ο δεύτερος σε σειρά είναι ο αλγόριθμος μηχανών διανυσμάτων υποστήριξης ο οποίος ανήκει στην κατηγορία επιβλεπόμενη μάθησης (supervised learning) και εφαρμόζετε σε προβλήματα classification και regression. Σκοπός του είναι να ταξινομήσει τις εξαρτημένες μεταβλητές σε κλάσεις σε έναν πολυδιάστατο χώρο (ο αριθμός των διαστάσεων εξαρτάται από τον αριθμό των ανεξάρτητων μεταβλητών) χρησιμοποιώντας διανύσματα.

Ανάμεσα σε πολλά διανύσματα ο αλγόριθμος προσπαθεί να βρει το βέλτιστο το οποίο θα διαχωρίζει τα δεδομένα στις κλάσεις.



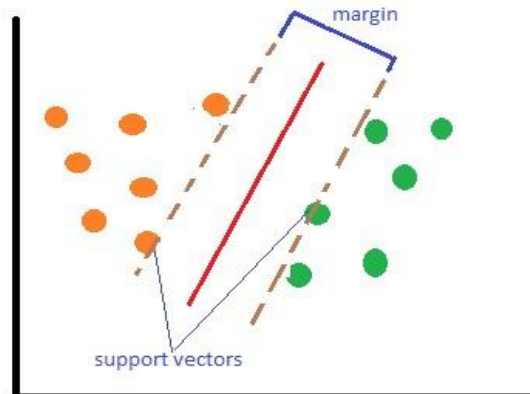
**Εικόνα 4.13** Βέλτιστο διάνυσμα.

Στην παραπάνω εικόνα απεικονίζονται δυο διανύσματα (A, B) σε δισδιάστατο χώρο τα οποία χωρίζουν τα δείγματα σε κλάσης. Όπως φαίνεται στην εικόνα το διάνυσμα A δεν ικανοποιεί ορθά το διαχωρισμό σε αντίθεση με το διάνυσμα B γιατί τα δείγματα είναι σε απόσταση από το διάνυσμα και είναι ταξινομημένα.

Η εξίσωση που περιγράφει το διάνυσμα B είναι η εξής:  $\mathbf{w} \cdot \mathbf{x} + \mathbf{b} = 0$ . Όπου w και b είναι σταθερές και το x είναι το δείγμα. Αφού τα δεδομένα είναι χωρισμένα η

## Ιατρικός φάκελος με δυνατότητα πρόβλεψης διαβήτη

απόσταση των κοντινότερων δειγμάτων που ανήκουν σε διαφορετικές κλάσης ονομάζεται margin (διακεκομμένη γραμμή) και είναι  $2/\|w\|$ .



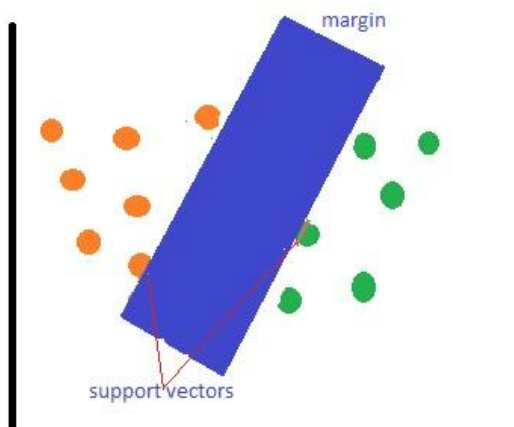
Εικόνα 4.14 Margin.

Έτσι όλα τα πορτοκαλή δείγματα που είναι πίσω από την διακεκομμένη γραμμή χαρακτηρίζονται ως 1 από την εξίσωση  $w \cdot x + b = 1$  και αντίστοιχα τα πράσινα με  $-1$   $w \cdot x + b = -1$ . Είναι σημαντικό το margin να είναι μέγιστο το δυνατόν για να γίνετε καλύτερος διαχωρισμός των δειγμάτων και το μεγιστοποιούμε χρησιμοποιώντας το Lagrange Multiplier.

$$\sum \alpha_i - \frac{1}{2} \sum_i \sum_j \alpha_i \alpha_j y_i y_j \vec{x}_i^T \vec{x}_j$$

Εικόνα 4.15 Lagrange Multiplier.

Το Lagrange Multiplier είναι πρόβλημα τετραγωνικού προγραμματισμού με σκοπό να βρει τις βέλτιστες τιμές του  $\alpha_i$  που μεγιστοποιούν τα όρια χωρίς όμως να υπάρχουν δείγματα εντός του margin.



Εικόνα 4.16 διαχωρισμός δεδομένων.

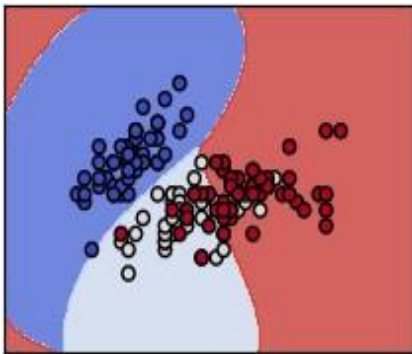


## Ιατρικός φάκελος με δυνατότητα πρόβλεψης διαβήτη

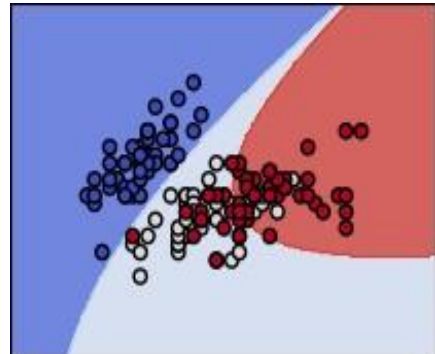
```
from sklearn import svm
start = time.time()
model= svm.SVC(kernel='linear', random_state=1)
model.fit(train,y_train)
predictions=model.predict(test)
print "%.3f " % accuracy_score(y_test, predictions)
predictions =model.predict([val])
print ",",predictions
end = time.time()
```

**Εικόνα 4.17** Μοντέλο του αλγορίθμου SVM.

Οι παράμετρος kernel δηλώνει τον τύπο της ταξινόμησης στην δικιά μας περίπτωση είναι γραμμικός υπάρχουν όμως και άλλοι τύποι όπως RBF και Polynomial. Τέλος για να έχουμε ντετερμινιστική έξοδο χρησιμοποιούμε μια σταθερά random\_state επειδή ο αλγόριθμος τείνει να αλλάζει ορισμένες τιμές οι οποίες επηρεάζουν την επίδοση του.



**Εικόνα 4.18**RBF kernel.

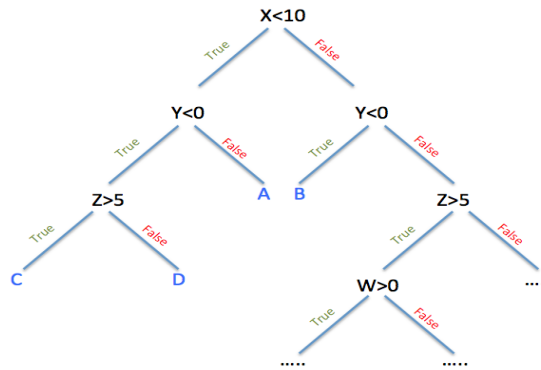


**Εικόνα 4.19**Polynomial kernel.

## Ιατρικός φάκελος με δυνατότητα πρόβλεψης διαβήτη

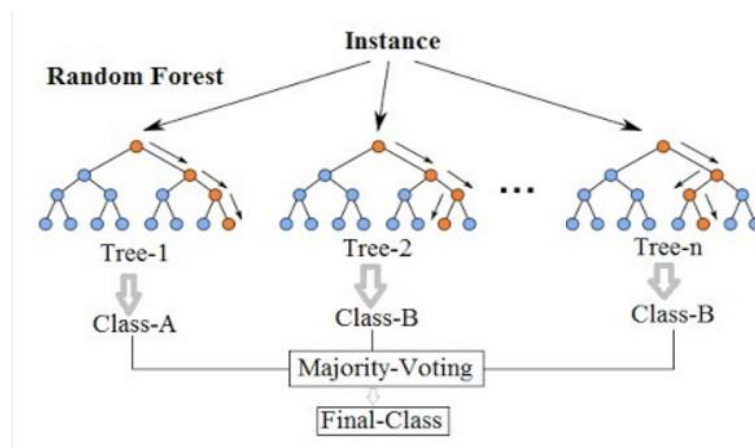
### 4.3.3 Random forest

Τέλος ο τελευταίος αλγόριθμος είναι ο random forest και η δομή του βασίζεται στα δυαδικά δένδρα. Το δένδρο χτίζεται παίρνοντας μια απόφαση (συνθήκη μεγαλύτερο ή μικρότερο) σε κάθε κόμβο μέχρι να φτάσει στο επιθυμητό αποτέλεσμα.



Εικόνα 4.20 Δένδρο αποφάσεων.

ο αλγόριθμος δημιουργεί  $N$  τέτοια δένδρα και το αποτέλεσμα (κλάση) κάθε δένδρου αποθηκεύεται και στο τέλος θα εμφανιστεί το αποτέλεσμα με το μεγαλύτερο αριθμό αποτελεσμάτων από τα δένδρα

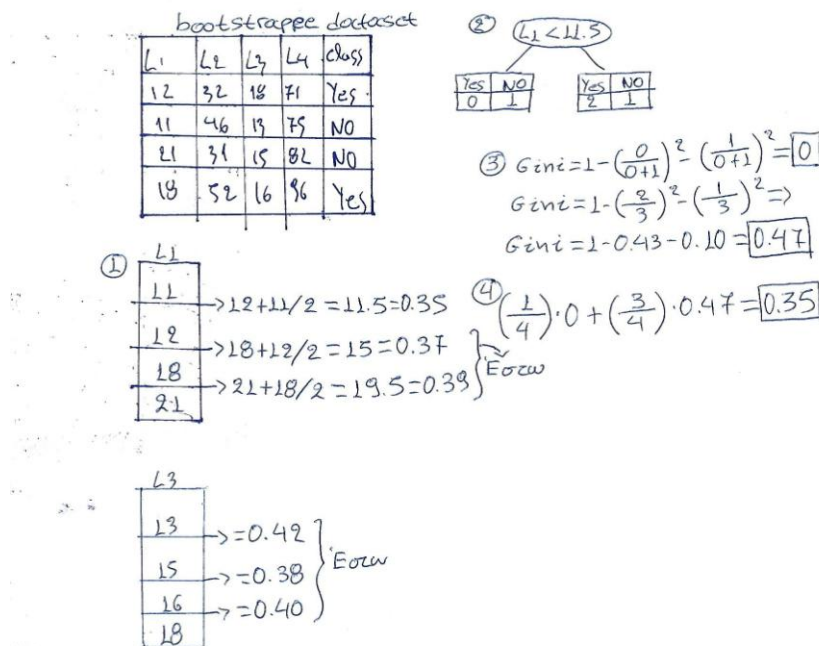


Εικόνα 4.21 Σύνολο δένδρων αποφάσεων.

Στην παραπάνω εικόνα δυο δένδρα πρόβλεψαν την κλάση B ενώ ένα πρόβλεψε A γι αυτό το αποτέλεσμα θα είναι η κλάση B επειδή την πρόβλεψαν περισσότερα δένδρα

## Ιατρικός φάκελος με δυνατότητα πρόβλεψης διαβήτη

Για να πάρει την απόφαση το δένδρο σε κάθε κόμβο ακολουθεί μια διαδικασία. Πρώτον, ο αλγόριθμος παίρνει τυχαία σειρές από τα δεδομένα και δημιουργεί μια bootstrapped dataset. Στη συνέχεια επιλέγει τυχαία από τη bootstrapped dataset στήλες και εφαρμόζει την εξίσωση Gini. Επιλέγει εκείνη τη στήλη με το μικρότερο αποτέλεσμα τις εξίσωσης. Έπειτα επιλέγει ξανά μια τυχαία στήλη και εφαρμόζει τις ίδιες μεθόδους έως ότου το δένδρο φτάσει στο αποτέλεσμα. Ύστερα κατασκευάζει το επόμενο δένδρο και φτιάχνει ξανά μια bootstrapped dataset.



Εικόνα 4.22 Μέθοδος Gini.

Από την παραπάνω εικόνα έστω ότι ο αλγόριθμος έχει επιλέξει τη στήλη L1 την οποία έχουμε ταξινομήσει. Έπειτα υπολογίζουμε την μέση τιμή των δειγμάτων και εφαρμόζουμε την εξίσωση Gini για όλες τις τιμές και επιλέγουμε το μικρότερο αποτέλεσμα. Στη συνέχεια ο αλγόριθμος επιλέγει άλλη στήλη (L3) και εφαρμόζει τις ίδιες μεθόδους και στο τέλος επιλέγει το μικρότερο αποτέλεσμα από τις δυο στήλες. Αφού ο αλγόριθμος έχει χτίσει το δένδρο θα χτίσει άλλα N δένδρα και στο τέλος θα εμφανίσει τη κλάση που θα την έχουν πρόβλεψη τα περισσότερα δένδρα

## Ιατρικός φάκελος με δυνατότητα πρόβλεψης διαβήτη

```
model= RandomForestClassifier(max_depth=18, random_state=34)
model.fit(train,y_train)
predictions=model.predict(test)
print "%.3f" % accuracy_score(y_test, predictions)
predictions =model.predict([val])
print ", ", predictions
```

**Εικόνα 4.23** Μοντέλο του αλγορίθμου RandomForest.

Η παράμετρος `max_depth` αντιπροσωπεύει το βάθος του κάθε δένδρου. Όσο μεγαλύτερο είναι το βάθος τόσο περισσότερη πληροφορία παίρνει για τα δεδομένα.

#### 4.3.4 Βιβλιοθήκες εγκατάστασης

- **Numpy**

Είναι βιβλιοθήκη για τη γλώσσα προγραμματισμού python και προσφέρει υποστήριξη για διαχωρισί πινάκων και για μήτρες.

- **Scipy**

Δημιουργήθηκε για τον επιστημονικό προγραμματισμό και πακέτα για γραμμική άλγεβρα, Fast Fourier Transform, επεξεργασία σηματον και εικόνας, optimization, integration.

- **Matplotlib**

Είναι πακέτο τις python και σχεδιάστηκε για την παραγωγή απλών και ισχυρών απεικονίσεων με ευκολία είναι.

- **Scikit-Learn**

Είναι πακέτο το οποίο περιέχει αλγορίθμους μηχανικής μάθησης και επεξεργασίας εικόνας.

- **Pandas**

Είναι πακέτο τις python και σχεδιάστηκε γρήγορο και εύκολο χειρισμό δεδομένων, συνάθροιση και οπτικοποίηση.

Η εγκατάσταση τις κάθε βιβλιοθήκης γίνεται πληκτρολογώντας την εντολή `pip install X` στο τερματικό όπου X το όνομα τις βιβλιοθήκης.

## ΚΕΦΑΛΑΙΟ 5

### Συμπεράσματα

Ύστερα από την ολοκλήρωση της εφαρμογής κατέληξα σε κάποια συμπεράσματα όσον αφορά τα Ιατρικά πληροφοριακά συστήματα και την μηχανική μάθηση

Τα Ιατρικά συστήματα αποτελούνται από πληθώρα παραμέτρους και πίνακες και πρέπει το μοντέλο τις βάσης να είναι καλά σχεδιασμένο πριν εφαρμοστεί. Έτσι ένα από τα προβλήματα που με δυσκόλεψαν είναι ο σχεδιασμός τις σχεσιακής βάσης. Επίσης άλλο πρόβλημα που αντιμετώπισα ήταν στο κομμάτι τις πρόβλεψης η ένωση τις ρyθση με την ρηρ διότι απαιτούσε αρκετή προσοχή και είχε διαδικασία να εφαρμοστεί από την ιδέα σε κώδικα Ένα άλλο απαιτητικό κομμάτι ήταν η εκμάθηση τον αλγορίθμων μηχανικής μάθησης διότι δεν έφτανε μόνο να μάθεις τι έκανε ο αλγόριθμος αλλά να μάθεις και το μαθηματικό του υπόβαθρο.

Κλείνοντας, η πτυχιακή μου έδωσε τη δυνατότητα να μάθω αρκετά πράγματα πάνω σε διάφορους κλάδους. Ο συνδυασμός τον Ιατρικών πληροφοριακών συστημάτων με την τεχνητή νοημοσύνη είναι μια σύγχρονη τεχνολογία πόσο μάλλον ο συνδυασμός Ιατρικής και τεχνητής νοημοσύνης ο τομέας αυτός είναι πολλά υποσχόμενος και έχει πολλά να δώσει και να διευκολύνει τις ζωές μας.

## ΒΙΒΛΙΟΓΡΑΦΙΑ

- [1] Παναγιωτης Δ. Κεντερλης “Αναπτυξη Διαδικτυακών Εφαρμογών θεωρητικά και πράξη” 2009.
- [2] Julie C. Meloni “PHP, MySQL και Apache” Τεταρτη εκδοση 2008.
- [3] Andrew S. Tanenbaum, David J. Wetherall “Δίκτυα Υπολογιστών” Πέμπτη Αμερικανική Εκδοση 2011.
- [4] Raghu Ramakrishnan, Johannes Gehrke “Συστήματα Διαχείρισης Βάσεων Δεδομένων” Τρίτη Εκδοση 2014.
- [5] Kim, Byung-Keun (2005). Internationalising the Internet the Co-evolution of Influence and Technology.
- [6] Segal, Ben (1995). "A Short History of Internet Protocols at CERN"
- [7] Berners-Lee, Tim. "The Original HTTP as defined in 1991"
- [8] Berners-Lee, Tim (c. 1993). "A Brief History of the Web". World Wide Web Consortium. Retrieved 17 August 2010
- [9] Baran, Paul (May 27, 1960). "Reliable Digital Communications Using Unreliable Network Repeater Nodes"
- [10] Technical Histories of the Internet & other Network Protocols. Computer Science Department, University of Texas Austin
- [11] Douglas E. Comer. Internetworking with TCP/IP – Principles, Protocols and Architecture.
- [12] Fielding, Roy T. Gettys, James; Mogul, Jeffrey C.; Nielsen, Henrik Frystyk Masinter, Larry; Leach, Paul J. Berners-Lee, Tim (June 1999).
- [13] Kurose, J. F.; Ross, K. W. (2010). Computer Networking: A Top-Down Approach
- [14] Domain Names - Concepts and Facilities, P. Mockapetris, The Internet Society (November 1987)
- [15] Stuart Russell, Peter Norvig “Τεχνητή Νοημοσύνη Μια σύγχρονη προσέγγιση” Δεύτερη Αμερικανική Εκδοση 2005.
- [16] Bowden, Margaret A. (2006), Mind As Machine: A History of Cognitive Science, Oxford University Press
- [17] Turing, Alan (October 1950), "Computing Machinery and Intelligence"
- [18] Schaeffer, Jonathan. One Jump Ahead Challenging Human Supremacy in Checkers.

## Ιατρικός φάκελος με δυνατότητα πρόβλεψης διαβήτη

- [19] Schmidhuber, J. (2015). "Deep Learning in Neural Networks: An Overview". Neural Networks.
- [20] Norvig, Peter (1992). Paradigms of artificial intelligence programming: case studies in Common Lisp.
- [21] Weizenbaum, Joseph (1976). Computer Power and Human Reason: From Judgment to Calculation.
- [22] Phil Simon (March 18, 2013). Too Big to Ignore: The Business Case for Big Data
- [23] Fayyad, Usama (1996). «From Data Mining to Knowledge Discovery in Databases»
- [24] Andrei Nikolajevich Kolmogorov (1950). Foundations of the Theory of Probability
- [25] Vapnik, Vladimir N. (1995). "Support-vector networks". Machine Learning