ΒΛΑΣΣΗΣ ΚΩΝΣΤΑΝΤΙΝΟΣ

236004

Θεωρία αποφάσεων

# Σκοπός

Στόχος της εργασίας είναι η δυνατότητα δημιουργίας ενός συστήματος πρόβλεψης για την παρουσία καρδιακής νόσου με χρήση ιστορικών δεδομένων. Χρησιμοποιούμε σαν ιστορικά δεδομένα ένα dataset που περιέχει 76 χαρακτηριστικά, αλλά όλα τα δηµοσιευµένα πειράµατα αναφέρονται στη χρήση ενός υποσυνόλου 14 από αυτά.

Τα χαρακτηριστικά είναι :

age, sex, cp, trestbps, chol, fbs, restecg, thalach, exang, oldpeak, slope, ca, thal

1. age

2. sex

3. chest pain type (4 values)

4. resting blood pressure

5. serum cholestoral in mg/dl

6. fasting blood sugar > 120 mg/dl

7. resting electrocardiographic results (values 0,1,2)

8. maximum heart rate achieved

9. exercise induced angina

10. oldpeak = ST depression induced by exercise relative to rest

11. the slope of the peak exercise ST segment

12. number of major vessels (0-3) colored by flourosopy

13. thal: 3 = normal; 6 = fixed defect; 7 = reversable defect

Ο στόχος είναι μια μεταβλητή που δίνεται 1 ή 0 έχει ή δεν έχει παρουσία νόσου.

# Θεωρία

Για την πρόβλεψη χρησιμοποιήθηκαν νευρωνικά δίκτυα και δέντρα αποφάσεων και συγκρίθηκαν μεταξύ τους.

Τα Νευρωνικά δίκτυα είναι συστήματα που μαθαίνουν μέσα από σειρά δεδομένων. Αποτελούν πλέον μια πολύ σημαντική λογική στις επιστήμες του Machine Lerning, τεχνικής νοημοσύνης καθώς ,Data Mining κλπ.

Τα Νευρωνικά δίκτυα μπορούν να χρησιμοποιηθούν ώστε να παράγονται μοντέλα ομαδοποίησης (classification) που με βάσει σειρά από στοιχεία αποφασίζουν.

Στην περίπτωση μας προτείνεται η λειτουργία ενός συστήματος όπου μια FPGA θα ελέγχει σειρά από διακόπτες (π.χ. φωτισμός σε ένα δωμάτιο, κουζίνα , θερμοσίφωνας , χρήση μπιστολιού κλπ). Το νευρωνικό δίκτυο αποτελεί το βασικό σύστημα μέσω του οποίου ανοίξει ένας άλλος διακόπτης.

Τα μοντέλα μηχανικής μάθησης λειτουργούν με την λογική «Μαθαίνω από τα δεδομένα». Έτσι στη περίπτωση μας έχουμε μια σειρά καταγραφών από το σύστημα μας. Κάθε μοντέλο μάθησης στόχο έχει να δημιουργήσει μια μηχανή που θα έχει το βέλτιστο αποτέλεσμα μεταξύ των τιμών που προβλέπει με αυτές που έχουν εκτιμηθεί από τους ειδικούς.

Έτσι σε κάθε μοντέλο υπάρχει ένα σύνολο μάθησης και ένα ή περισσότερα σύνολα τεστ με τα οποία ελέγχουμε πόσο το σύστημα μας έχει «μάθει σωστά». Τα μοντέλα αυτά παράγονται έχοντας σαν είσοδο τους παράγοντες μαζί με τις εκτιμήσεις.

Στην συνέχεια κάθε μοντέλο ακολουθεί μια μεθοδολογία με στόχο να καταλήξει στο σημείο τερματισμού, που για κάθε μοντέλο είναι το μεγαλύτερο δυνατό για αυτό ποσοστό πρόβλεψης.

H μηχανική μάθηση δεν απαιτεί προηγούμενες παραδοχές σχετικά με τις σχέσεις μεταξύ των μεταβλητών. Στα μοντέλα της μηχανικής μάθησης πρέπει απλά να δώσουμε όλα τα δεδομένα που έχουμε συλλέξει, και ο αλγόριθμος επεξεργάζεται τα δεδομένα και ανακαλύπτει τα πρότυπα, με τα οποία μπορούμε να κάνουμε προβλέψεις για το νέο σύνολο δεδομένων (τεστ σετ). Η μηχανική μάθηση αντιμετωπίζει έναν αλγόριθμο **σαν ένα μαύρο κουτί (black box),** για όσο διάστημα αυτό λειτουργεί. Γενικά εφαρμόζεται σε σύνολα με πολλές διαστάσεις (δηλαδή πολλές μεταβλητές) ή σε σύνολα με πολλές καταγραφές (παρατηρήσεις), όσο περισσότερα είναι τα δεδομένα, τόσο πιο ακριβή θα είναι η πρόβλεψή του μοντέλου.

Δηλαδή η μηχανική μάθηση λειτουργεί με τρόπο διαφορετικό με την στατιστική. Στη στατιστική τα μοντέλα που αναζητούνται απαιτούν να ξέρει κάποιος πώς συλλέχθηκαν τα δεδομένα, τις στατιστικές ιδιότητες των εκτιμητών ( p-value, αμερόληπτες εκτιμήτριες κ.α.), την υποκείμενη κατανομή του πληθυσμού που μελετούμε π.χ. στην περίπτωση μας τις καταγραφές κατολισθήσεων, και τα είδη των ιδιοτήτων για κάθε μεταβλητή. Θα πρέπει να ξέρει ακριβώς τι κάνει για να καταλήξει σε παραμέτρους που θα παρέχουν την ικανότητα πρόβλεψης. Οι τεχνικές στατιστικής μοντελοποίησης εφαρμόζονται συνήθως σε σύνολα χαμηλών διαστάσεων των δεδομένων, δηλαδή λιγότερες μεταβλητές και παρατηρήσεις.

Φυσικά και τα μοντέλα μηχανικής μάθησης χρησιμοποιούν στατιστική μέχρι να καταλήξουν στα αποτελέσματα τους, παρόλα αυτά το τελικό μοντέλο είναι κάτι αυτόνομο και πολλές φορές η φυσική σημασία δεν έχει ακριβώς τόση έννοια αφού ουσιαστικά προσπαθούν να αποδώσουν την σύνθεση των παραγόντων μεταξύ τους με τέτοιό τρόπο που απλά να ικανοποιεί τα δεδομένα και τα αποτελέσματα αυτών. Ειδικά σε πολυπαραγοντικά συστήματα τα μοντέλα αυτά δίνουν πολλαπλές σχέσεις μεταξύ των παραγόντων με τέτοιο τρόπο που είναι σχεδόν αδύνατον να καταλάβει κανείς το γιατί. Λειτουργούν δηλαδή σαν ένα απλό φυσικό σύστημα που μαθαίνει, όπως ένα παιδί που μεγαλώνει μαθαίνει να κάνει πράξεις . Ο τρόπος που οι νευρώνες λειτουργούν μεταξύ τους ώστε να δοθεί το αποτέλεσμα είναι πρακτικά ακατανόητος. (Shai Shalev-Shwartz and Shai Ben-David , Understanding Machine Learning: From

Το νευρωνικό δίκτυο είναι ένα δίκτυο από απλούς υπολογιστικούς κόμβους (νευρώνες, νευρώνια), διασυνδεδεμένους μεταξύ τους. Είναι εμπνευσμένο από το Κεντρικό Νευρικό Σύστημα (ΚΝΣ), το οποίο προσπαθεί να προσομοιώσει.

Οι νευρώνες είναι τα δομικά στοιχεία του δικτύου. Κάθε τέτοιος κόμβος δέχεται ένα σύνολο αριθμητικών εισόδων από διαφορετικές πηγές (είτε από άλλους νευρώνες, είτε από το περιβάλλον), επιτελεί έναν υπολογισμό με βάση αυτές τις εισόδους και παράγει μία έξοδο. Η εν λόγω έξοδος είτε κατευθύνεται στο περιβάλλον, είτε τροφοδοτείται ως είσοδος σε άλλους νευρώνες του δικτύου. Υπάρχουν τρεις τύποι νευρώνων: οι νευρώνες εισόδου, οι νευρώνες εξόδου και οι υπολογιστικοί νευρώνες ή κρυμμένοι νευρώνες. Οι νευρώνες εισόδου δεν επιτελούν κανέναν υπολογισμό, μεσολαβούν απλώς ανάμεσα στις περιβαλλοντικές εισόδους του δικτύου και στους υπολογιστικούς νευρώνες. Οι νευρώνες εξόδου διοχετεύουν στο περιβάλλον τις τελικές αριθμητικές εξόδους του δικτύου. Οι υπολογιστικοί νευρώνες πολλαπλασιάζουν κάθε είσοδό τους με το αντίστοιχο συναπτικό βάρος και υπολογίζουν το ολικό άθροισμα των γινομένων. Το άθροισμα αυτό τροφοδοτείται ως όρισμα στη συνάρτηση ενεργοποίησης, την οποία υλοποιεί εσωτερικά κάθε κόμβος. Η τιμή που λαμβάνει η συνάρτηση για το εν λόγω όρισμα είναι και η έξοδος του νευρώνα για τις τρέχουσες εισόδους και βάρη.

Τα δέντρα απόφασης χρησιμοποιούνται ευρέως για την κατηγοριοποίηση και πρόβλεψη δεδομένων. Ένα δέντρο απόφασης κατασκευάζεται σύμφωνα με ένα σύνολο εκπαίδευσης προ-κατηγοριοποιημένων δεδομένων. Κάθε εσωτερικός κόμβος προσδιορίζει τον έλεγχο των γνωρισμάτων και κάθε κλαδί που συνδέει τους εσωτερικούς με τους απόγονους αντιστοιχεί σε μία πιθανή τιμή για το γνώρισμα.

Υπάρχουν πολλοί αλγόριθμοι για την δημιουργία του καλύτερου δέντρου απόφασης με γνωστότερο τον ID3 (Iterative Dichotomiser 3)

# Yλοποίηση και αποτελέσματα

Δημιουργήθηκε ο παρακάτω κώδικας που αφορά την πρόβλεψη με δέντρα απόφασης αλλά και 4 νευρώνικα δίκτυα με διαφορετικά κρυφά επίπεδα

Ο κώδικας της υλοποίησης είναι ο παρακάτω:

|  |
| --- |
| import pandas as pd  import numpy as np  from sklearn import tree  from sklearn.tree import DecisionTreeClassifier  from sklearn.neural\_network import MLPClassifier  from sklearn.model\_selection import train\_test\_split  df=pd.read\_csv("Dataset 1.csv")  features = ['age','sex','cp','trestbps','chol','fbs','restecg','thalach','exang','oldpeak','slope','ca','thal']  X = df[features]  y = df['target']  X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.33, random\_state=42)  dtree = DecisionTreeClassifier()  dtree = dtree.fit(X\_train, y\_train)  print("Score of Tree:")  print(dtree.score(X\_test,y\_test))  a=len(features)  nn = MLPClassifier(random\_state=0, max\_iter=500,hidden\_layer\_sizes=(a,)).fit(X\_train,y\_train)  dtree = nn.fit(X\_train, y\_train)  print("Score of Neural Networks1:")  print(nn.score(X\_test,y\_test))  nn = MLPClassifier(random\_state=0, max\_iter=500,hidden\_layer\_sizes=(2\*a,a)).fit(X\_train,y\_train)  dtree = nn.fit(X\_train, y\_train)  print("Score of Neural Networks1:")  print(nn.score(X\_test,y\_test))  nn = MLPClassifier(random\_state=0, max\_iter=500,hidden\_layer\_sizes=(2\*a,a,int(a/2))).fit(X\_train,y\_train)  dtree = nn.fit(X\_train, y\_train)  print("Score of Neural Networks2:")  print(nn.score(X\_test,y\_test))  nn = MLPClassifier(random\_state=0, max\_iter=500,hidden\_layer\_sizes=(3\*a,2\*a,a)).fit(X\_train,y\_train)  dtree = nn.fit(X\_train, y\_train)  print("Score of Neural Networks3:")  print(nn.score(X\_test,y\_test))  nn = MLPClassifier(random\_state=0).fit(X\_train,y\_train)  dtree = nn.fit(X\_train, y\_train)  print("Score of Neural Networks3:")  print(nn.score(X\_test,y\_test)) |

Η διαδικασία ήταν να χωρίσουμε το σύνολο σε Training και Test (33% το Test) με τυχαίο τρόπο και να εφαρμόσουμε τα διαφορετικά μοντέλα μας

Τελικα πήραμε τα παρακάτω αποτελέσματα

Score of Tree: 0.74

Score of Neural Networks1: 0.63

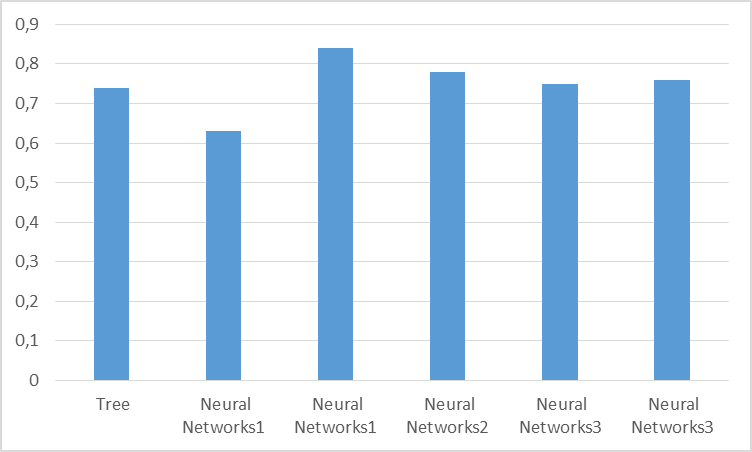
Score of Neural Networks1: 0.84

Score of Neural Networks2: 0.78

Score of Neural Networks3: 0.75

Score of Neural Networks3: 0.76

Γραφικά έχουμε το παρακάτω:



Όπως βλέπουμε όλα τα μοντέλα δίνουν αποτέλεσμα πάνω από 65% ενώ το νευρωνικό με 2 κρυφά επίπεδα με πλήθος νευρώνων 2\*πλήθος χαρακτηριστικών στο πρώτο επίπεδο και πλήθος χαρακτηριστικών στο δεύτερο δίνει τη καλύτερη πρόβλεψη με 84% ακρίβεια.

Επίσης για επιβεβαίωση των κλάσεων πήραμε και τα παρακάτω μέτρα που επιβεβαιώνουν την πιο σωστή πρόβλεψη με το καλύτερο νευρωνικό

**precision recall f1-score support**

0 0.79 0.82 0.80 40

1 0.88 0.85 0.86 60

accuracy 0.84 100

macro avg 0.83 0.84 0.83 100

weighted avg 0.84 0.84 0.84 100