

Κοκολάκη Στέλλα Π2016177

Σαϊλάκης Κωσταντίνος Π2016121

Επιβλέπον: Θεμιστοκλής Έξαρχος

Συστήματα Υποστήριξης Απόφασης

### Σύστημα Υποστήριξης Απόφασης για την Ηπατίτιδα

Στα πλαίσια του μαθήματος δημιουργήσαμε ένα dss με Weka και python. Για την αρχική ανάλυση των δεδομένων και την επιλογή αλγορίθμου του συστήματος υποστήριξης απόφασης εργαστήκαμε με Weka και έπειτα υλοποιήσαμε το σύστημα υποστήριξης απόφασης με python.

Με τη βοήθεια του Weka επιλέξαμε για τα δικά μας δεδομένα τον αλγόριθμο με τη καλύτερη ακρίβεια ανάμεσα στους αλγορίθμους j48, random forest, k-NN και naive Bayes. Η επιλογή μας έγινε σύμφωνα με τις τιμές των αλγορίθμων στις κατηγορίες Accuracy, TP Rate, FP Rate, Precision ,Recall, F-Measure. Εξαιτίας των κοντινών τιμών που είχαν οι αλγόριθμοι στα πειράματά μας, συμβουλευτήκαμε ιδιαίτερα την τιμή της ακρίβειας F-Measure.

Τέλος διαπιστώσαμε ότι ο Naive Bayes είχε την καλύτερη απόδοση και υλοποιήσαμε την τελική προετοιμασία των δεδομένων αλλά και την εφαρμογή μας με python με αυτόν.

Ο κώδικας της εφαρμογής είναι στο παρακάτω αποθετήριο στο github:

<https://github.com/stellakokolaki/Hepatitis-DSS->

Για το μάθημα υλοποιήθηκε εφαρμογή για Mac OS όπου και παρουσιάστηκε στα πλαίσια του μαθήματος. Στον σύνδεσμο θα βρείτε τον κώδικα της εφαρμογής με εκτέλεση του απο το τερματικό με python3.

## Προετοιμασία Δεδομένων

Για να εισαγάγουμε το αρχείο arff στην Python χρησιμοποιήσαμε την συνάρτηση `io.arff.loadarff` του πακέτου `scipy`. Αυτή η συνάρτηση μας επιτρέπει να φορτώσουμε το dataset σε ένα Pandas DataFrame. Το DataFrame είναι μία δομή δεδομένων που έχει δημιουργηθεί για επεξεργασία μεγάλων dataset από την βιβλιοθήκη `Pandas` και προσφέρει πολλές έτοιμες συναρτήσεις.

Το πρώτο πρόβλημα που αντιμετωπίζουμε είναι ότι η συνάρτηση `io.arff.loadarff` δεν μετατρέπει απευθείας τα strings του dataset σε αντικείμενα string της Python. Αντιθέτως τα εισαγάγει ως απλά bytes τα οποία δεν μπορούμε να επεξεργαστούμε απευθείας. Αυτό φαίνεται στην παρακάτω εικόνα:

Για να επιλύσουμε αυτό το πρόβλημα χρησιμοποιούμε την συνάρτηση `decode` της Python η οποία δέχεται ένα byte object και το μετατρέπει σε utf-8 string. Εφαρμόζουμε επαναληπτικά την συνάρτηση `decode` σε κάθε τιμή των στηλών 1 έως 12, δηλαδή SEX έως VARICES, καθώς και στις στήλες 18, 19 αφού και αυτές έχουν δεδομένα string.

Αφού τρέξουμε τον παραπάνω κώδικα και εκτυπώσουμε τα δεδομένα, διαπιστώνουμε το δεύτερο πρόβλημα που πρέπει να αντιμετωπίσουμε. Αυτό φαίνεται στην παρακάτω εικόνα:

Για να αντιμετωπίσουμε αυτό το πρόβλημα δημιουργούμε την συνάρτηση `str_repl`.

```

25  ### REPLACE ? WITH OTHER STRING
26  def str_repl(str, weights = None):
27      if str == '?':
28          return random.choices(["yes", "no"], weights = weights)[0]
29      else:
30          return str
31  
```

150	46.0	female	yes	no	yes	yes	yes	yes	no	no
151	44.0	female	yes	no	yes	no	no	yes	yes	no
152	61.0	female	no	no	yes	yes	no	no	yes	no
153	53.0	male	no	no	yes	no	no	yes	no	yes
154	43.0	female	yes	no	yes	no	no	yes	no	yes

155 rows x 20 columns

155 rows x 20 columns

Χρησιμοποιούμε την συνάρτηση `str_repl` σε κάθε στήλη που έχουμε τιμές string και για κάθε στήλη χρησιμοποιούμε τις κατάλληλες πιθανότητες για “yes” και “no” σύμφωνα με τον αριθμό εμφάνισής τους.

```

32  for i in range(2,13):
33      row = df.iloc[:, i]
34      y = sum(row.str.count("yes"))
35      n = sum(row.str.count("no"))
36      weights = [y/(y+n), n/(y+n)]
37      df.iloc[:, i] = df.iloc[:, i].apply(lambda x : str_repl(x, weights = weights), 1)
38
39  row = df.iloc[:, 18]
40  y = sum(row.str.count("yes"))
41  n = sum(row.str.count("no"))
42  weights = [y/(y+n), n/(y+n)]
43  df.iloc[:, 18] = df.iloc[:, 18].apply(lambda x : str_repl(x, weights = weights), 1)
44  
```

## Δημιουργία Διεπαφής

Για την δημιουργία της διεπαφής χρησιμοποιήσαμε την βιβλιοθήκη `tinter`. Δημιουργήσαμε ένα απλό παράθυρο που περιέχει ένα πεδίο συμπλήρωσης κειμένου για καθένα από τα χαρακτηριστικά των δεδομένων εκτός από την κλάση πρόβλεψης. Στο τέλος του παραθύρου βρίσκεται το κουμπί Submit το οποίο όταν πατηθεί εξαγαγάγει τις τιμές από τα πεδία κειμένου και ξεκινά την διαδικασία πρόβλεψης της κλάσης.

Αυτή η διαδικασία χρησιμοποιεί την συνάρτηση `LabelBinarizer` για να μετατρέψουμε τα string labels σε αριθμούς 0 ή 1. Έπειτα, εκπαιδεύουμε έναν `NaiveBayesClassifier` με τον 67% των δεδομένων και προβλέπουμε την κλάση του παραδείγματος που μας δόθηκε. Τέλος εμφανίζουμε ένα μήνυμα (popup) με το αποτέλεσμα στο παράθυρο της διεπαφής.

## Τιμές που γίνονται δεκτές απο το DSS


<b>Age</b>	ακέραιος αριθμός
<b>Sex</b>	male, female
<b>Steroid</b>	yes, no
<b>Antivirals</b>	yes, no
<b>Fatigue</b>	yes, no
<b>Malaise</b>	yes, no
<b>Anorexia</b>	yes, no
<b>Liver Big</b>	yes, no
<b>Liver Firm</b>	yes, no
<b>Spleen Palpable</b>	yes, no
<b>Spiders</b>	yes, no
<b>Ascites</b>	yes, no
<b>Verices</b>	yes, no
<b>Bilirubin</b>	ακέραιος/δεκαδικός αριθμός
<b>Alk Phosphate</b>	ακέραιος αριθμός
<b>SGOT</b>	ακέραιος αριθμός
<b>Albumin</b>	ακέραιος/δεκαδικός αριθμός
<b>Protime</b>	ακέραιος/δεκαδικός αριθμός
<b>Histology</b>	yes, no

## Γραφικό Περιβάλλον Εφαρμογής

Welcome to this DSS

Age	30
Sex	male
Steroid	no
Antivirals	no
Fatigue	no
Malaise	no
Anorexia	no
Liver Big	no
Liver Firm	no
Spleen Palpable	no
Spiders	no
Ascites	no
Varices	no
Bilirubin	0.7
Alk Phosphate	85
SGOT	18
Albumin	4
Protime	80
Histology	no

Hepatitis Prediction:

 **Live**