Αναφορά Project για το μάθημα «Εξόρυξη Δεδομένων» - Ακαδημαϊκό Έτος 2020 – 2021

**Ομάδα**

**Καρδούτσος Άγγελος - Α.Μ.: 1059372, Έτος: 4ο, Τμήμα: ΗΜΤΥ**

**Παπαδόπουλος Ελευθέριος - Α.Μ.: 105ΧΧΧΧ, Έτος: 4ο, Τμήμα: ΗΜΤΥ**

Περιεχόμενα

[1. Εισαγωγή 1](#_Toc74863264)

[2. Επίλυση Ασκήσεων 2](#_Toc74863265)

[2.1 Άσκηση 1 2](#_Toc74863266)

[2.1.1 Προ-επεξεργασία 3](#_Toc74863267)

[2.1.2 Ερώτημα Α 3](#_Toc74863268)

[2.1.3 Ερώτημα Β 4](#_Toc74863269)

[2.1.4 Ερώτημα Γ 7](#_Toc74863270)

[2.2 Άσκηση 2 9](#_Toc74863271)

# 1. Εισαγωγή

Στόχος της συγκεκριμένης εργασίας αποτελεί η επίλυση δύο προβλημάτων κατηγοριοποίησης. Ειδικότερα, σε κάθε πρόβλημα, έχουμε στη διάθεση μας ένα αρχείο csv, το οποίο περιέχει παραδείγματα για όλες τις κλάσεις του προβλήματος. Ένα παράδειγμα χαρακτηρίζεται από γνωρίσματα, των οποίων οι τιμές μπορεί να είναι συνεχείς, διακριτές ή αλφαριθμητικές. Επίσης, διαθέτει και την κλάση/κατηγορία, στην οποία ανήκει το παράδειγμα. Σκοπός μας είναι να μελετήσουμε τα γνωρίσματα των παραδειγμάτων και τις τιμές τους, να δημιουργήσουμε ένα μοντέλο, το οποίο ταξινομεί κάθε παράδειγμα σε μία κλάση και, τέλος, να αξιολογήσουμε την ακρίβεια του μοντέλου μας.

Τα προγράμματα των ασκήσεων 1 και 2 αναπτύχθηκαν με γλώσσα προγραμματισμού Python. Επιπλέον, αξιοποιήθηκαν συναρτήσεις και δομές των βιβλιοθηκών:

* pandas
* numpy
* sklearn
* keras
* seaborn
* matplotlib

# 2. Επίλυση Ασκήσεων

Για λόγους οργάνωσης, για κάθε μία από τις 2 ασκήσεις, δημιουργήθηκε ξεχωριστό αρχείο .py, το οποίο συνοδεύεται από ένα αρχείο .csv στον ίδιο φάκελο. Συγκεκριμένα, για την άσκηση 1, το πρόγραμμα βρίσκεται στο αρχείο *ex1\_v1.py*, ενώ για την άσκηση 2, το πρόγραμμα βρίσκεται στο αρχείο *Ex2.py*.

## 2.1 Άσκηση 1

Στο αρχείο *healthcare-dataset-stroke-data.csv* εντοπίζονται δεδομένα για 5110 ασθενείς, όπου κάθε γραμμή αντιστοιχεί σε πληροφορίες για ένα ασθενή, κάθε τιμή σε μια γραμμή αντιστοιχεί στην τιμή ενός συγκεκριμένου γνωρίσματος, ενώ η τελευταία τιμή αντιστοιχεί στην κλάση στην οποία ανήκει το συγκεκριμένο παράδειγμα/ασθενής. Οι κλάσεις είναι 2 και έχουν τιμές:

* 1, αν ο ασθενής έχει υποστεί εγκεφαλικό επεισόδιο.
* 0, αν ο ασθενής δεν έχει υποστεί εγκεφαλικό επεισόδιο.

Παρακάτω παρατίθενται και τα γνωρίσματα των παραδειγμάτων, χωρισμένα σε τρεις κατηγορίες:

1. Γνωρίσματα με διακριτές τιμές:
   1. id: κωδικός ασθενούς
   2. age: ηλικία
   3. hypertension: ο ασθενής έχει (1) ή δεν έχει (0) υπέρταση
   4. heart\_disease: ο ασθενής έχει (1) ή δεν έχει (0) κάποια καρδιακή πάθηση
2. Γνωρίσματα με συνεχείς τιμές:
   1. avg\_glucose: μέσα επίπεδα γλυκόζης
   2. bmi: τιμή του δείκτη BMI του ασθενούς
3. Γνωρίσματα με αλφαριθμητικές τιμές:
   1. gender: φύλο ασθενούς (“Male”, “Female”, “Other”)
   2. ever\_married: έγγαμη κατάσταση (“Yes”, “No”)
   3. work\_type: τύπος εργασίας ("children", "Govt\_jov", "Never\_worked", "Private", "Self-employed")
   4. Residence: τοποθεσία κατοικίας (“Rural”, “Urban”)
   5. smoking\_status: καπνιστικές συνήθειες (“formerly smoked”, “never smoked”, “smokes”, “Unknown)

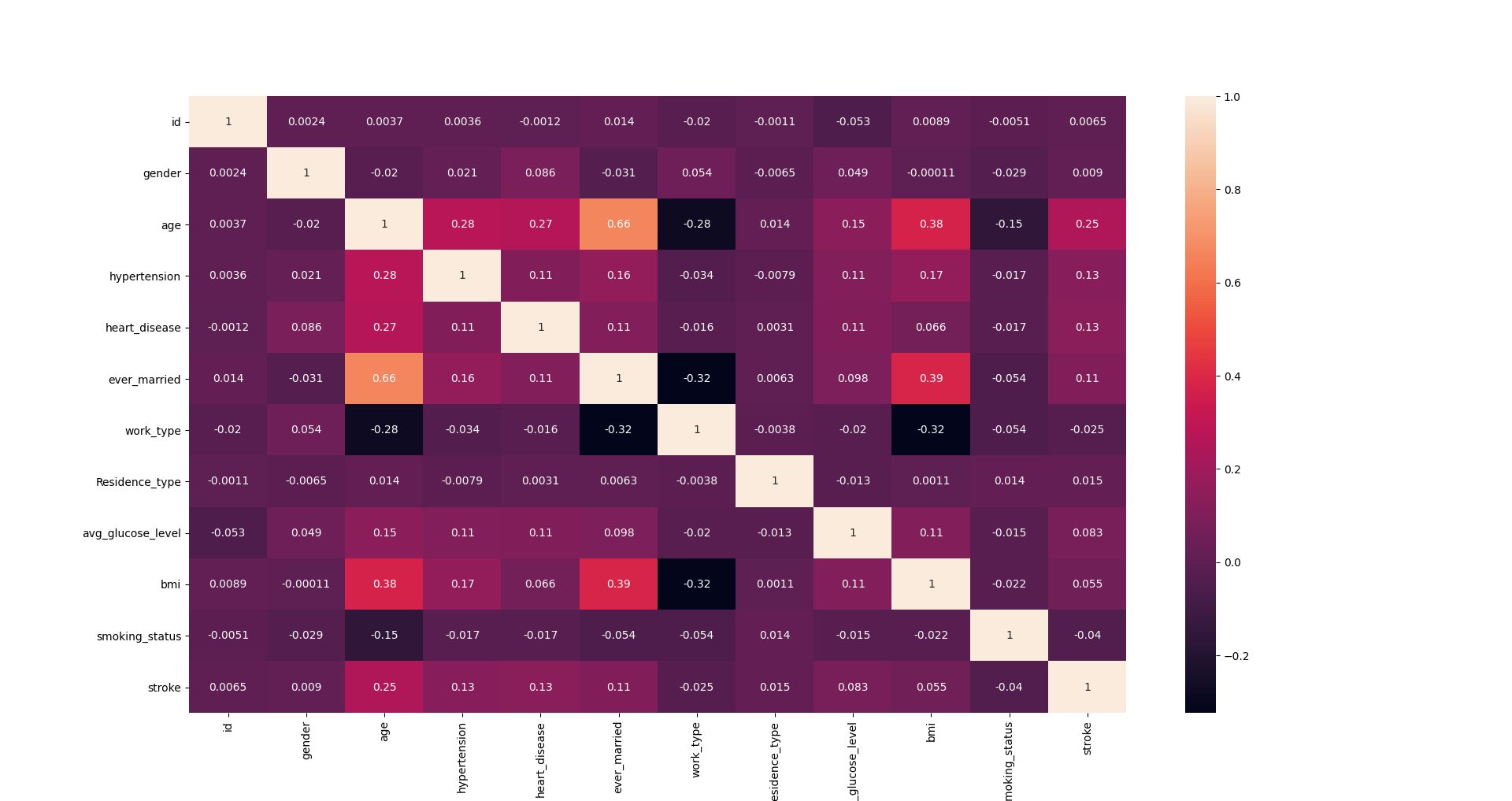
Τρέχοντας το πρόγραμμα, εμφανίζεται στο τερματικό ένα μενού από το οποίο μπορούμε να επιλέξουμε να εκτελεστεί το τμήμα του κώδικα που αντιστοιχεί σε ένα συγκεκριμένο ερώτημα της άσκησης. Ειδικότερα, για το ερώτημα Β, μπορούμε να επιλέξουμε ένα από τα τέσσερα υπο-ερωτήματα για να εκτελεστεί. Αφού πραγματοποιήσουμε την επιλογή μας και εκτελεστεί ο κώδικας, τα αποτελέσματα εμφανίζονται στο τερματικό ή ανοίγει ένα δεύτερο παράθυρο, το οποίο περιέχει κάποια γραφική παράσταση.

### 2.1.1 Προ-επεξεργασία

Αρχικά, διαβάζουμε και αποθηκεύουμε τα δεδομένα του csv αρχείου σε ένα Dataframe, με την εντολή ***read\_csv*** της βιβλιοθήκης. Στη συνέχεια, πρέπει να μετατρέψουμε τα αλφαριθμητικά δεδομένα σε διακριτές τιμές, ώστε να είναι δυνατή η αξιοποίησή τους σε επόμενα ερωτήματα. Για το λόγο αυτό δημιουργούμε τη συνάρτηση ***catCodes***, η οποία επιτελεί τη συγκεκριμένη εργασία.

### 2.1.2 Ερώτημα Α

Στα πλαίσια αυτού του ερωτήματος δημιουργήθηκε ένα heatmap, το οποίο παρουσιάζει τη συσχέτιση κατά Spearman μεταξύ των γνωρισμάτων των παραδειγμάτων. Για τη δημιουργία του, χρησιμοποιήθηκαν συναρτήσεις των βιβλιοθηκών ***pandas***, ***seaborn*** και ***matplotlib***, ενώ παράλληλα δημιουργήσαμε τη συνάρτηση ***exerciseA***.

**Αποτελέσματα:**

Εικόνα : Heatmap συσχετίσεων των γνωρισμάτων

Παρατηρούμε ότι συγκεκριμένα γνωρίσματα όπως το id και το φύλο είναι ασυσχέτιστα ως προς τα άλλα γνωρίσματα, καθώς οι τιμές των συσχετίσεων είναι κοντά στο 0. Παράλληλα υπάρχουν ορισμένα γνωρίσματα που παρουσιάζουν ισχυρότερη συσχέτιση με άλλα γνωρίσματα, όπως για παράδειγμα η ηλικία και το bmi, καθώς η τιμή τους πλησιάζει προς το 1 (ή το -1).

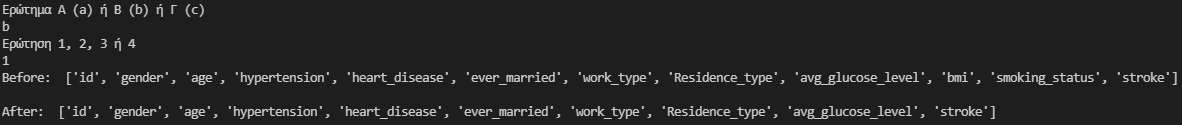
### 2.1.3 Ερώτημα Β

Παρατηρούμε ότι σε ορισμένα παραδείγματα λείπουν οι τιμές συγκεκριμένων γνωρισμάτων. Ειδικότερα, σε 201 παραδείγματα η τιμή του “BMI” είναι κενή (NaN), ενώ σε 1544 παραδείγματα η τιμή του “Smoking Status” είναι “Unknown”, δηλαδή δε γνωρίζουμε αν ο ασθενής κάπνιζε ή όχι, οπότε μπορούμε να την θεωρήσουμε ελλιπή τιμή. Στα ακόλουθα τέσσερα υπο-ερωτήματα θα εφαρμόσουμε διάφορες μεθόδους και αλγόριθμους για να συμπληρώσουμε αυτές τις ελλιπείς τιμές.

#### Ερώτημα Β1

Η πρώτη μέθοδος και πιο απλή που εφαρμόζουμε είναι να αφαιρέσουμε τις στήλες “bmi” και “smoking\_status” από το dataframe με χρήση της μεθόδου ***drop***.

Στο τερματικό εμφανίζουμε τη λίστα των γνωρισμάτων που υπάρχουν στο dataframe πριν και μετά την αφαίρεση των 2 γνωρισμάτων.

**Αποτελέσματα:**

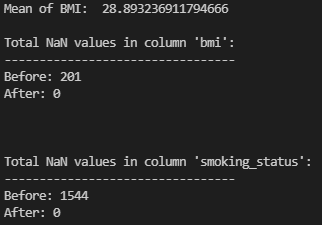
Εικόνα : Γνωρίσματα στο dataframe πριν και μετά το drop

#### Ερώτημα Β2

Σε αυτή την περίπτωση επιλέγουμε να μην διαγράψουμε τις στήλες, αλλά να συμπληρώσουμε τις ελλιπείς τιμές. Δημιουργούμε αρχικά τη συνάρτηση ***exerciseB\_2***. Για το “bmi” υπολογίζουμε το μέσο όρο των τιμών του γνωρίσματος από τα άλλα παραδείγματα και στη συνέχεια συμπληρώνουμε τις κενές τιμές με τη μέθοδο ***fillna***. Αντιθέτως, για το “smoking status”, δεν μπορούμε να υπολογίσουμε το μέσο όρο των διακριτών τιμών που θέσαμε, καθώς οι τιμές αυτές δεν έχουν κάποια φυσική σημασία, αλλά αντιστοιχούν σε αλφαριθμητικά. Επομένως, επιλέγουμε να συμπληρώσουμε τις ελλιπείς τιμές με εκείνη την τιμή που έχει τη μεγαλύτερη συχνότητα εμφάνισης στο γνώρισμα.

Στο τερματικό εμφανίζουμε τη μέση τιμή “bmi”, καθώς και το πλήθος των NaN τιμών για κάθε γνώρισμα, πριν και μετά το γέμισμα.

**Αποτελέσματα:**



Εικόνα :Πλήθος NaN τιμών πριν και μετά το fillna

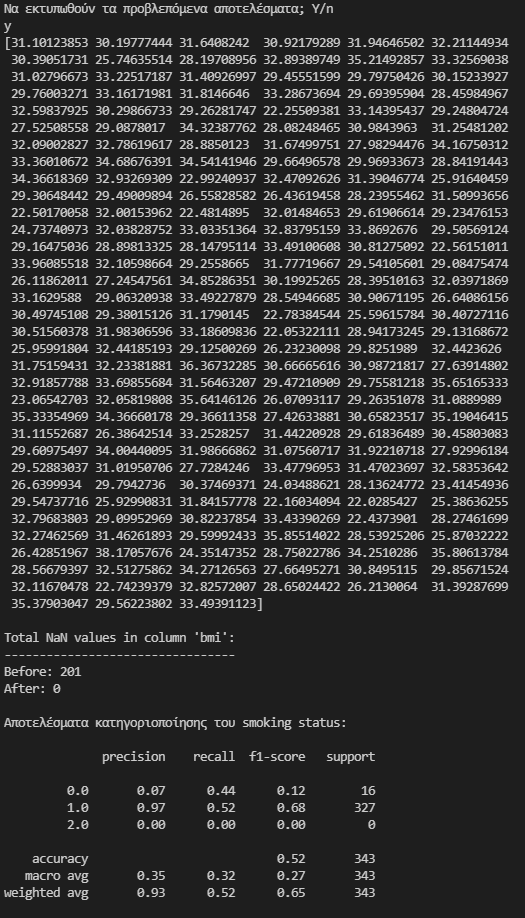
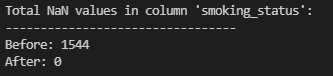
#### Ερώτημα Β3

Σε αυτό το ερώτημα αξιοποιούμε το γραμμικό μοντέλο ***LinearRegression*** της βιβλιοθήκης ***sklearn***. Δημιουργούμε αρχικά τη συνάρτηση exerciseB\_3, εντός της οποίας δημιουργούμε, αρχικά, ένα στιγμιότυπο της κλάσης ***LinearRegression*** με σκοπό να υπολογίσουμε τις NaN τιμές του “bmi”. Στη συνέχεια, διαχωρίζουμε το dataframe σε δύο dataframes. Το πρώτο περιέχει όλα τα παραδείγματα, εκτός αυτών που έχουν ελλιπείς τιμές, χωρίς να περιλαμβάνει τη στήλη “bmi”, λόγω των ελλιπών τιμών της, τη στήλη “stroke”, καθώς αποτελεί την κλάση και τη στήλη “id”, η οποία δεν περιέχει κάποια ουσιαστική πληροφορία. Στο δεύτερο τοποθετούμε όλα τα παραδείγματα, εκτός αυτών που έχουν ελλιπείς τιμές, και περιλαμβάνουμε μόνο τη στήλη “bmi”. Διαχωρίζουμε τα ανωτέρω δεδομένα σε dataframes για εκπαίδευση και dataframes για τα tests με αναλογία 90%-10%. Εκπαιδεύουμε το μοντέλο καλώντας τη μέθοδο ***fit*** με τα training data. Τέλος, δημιουργούμε ένα dataframe που περιέχει όλα τα παραδείγματα με ελλιπείς τιμές στο “bmi” και χρησιμοποιούμε για προβλέψουμε τις τιμές αυτές. Συμπληρώνουμε τις προβλεπόμενες τιμές στο αρχικό μας dataframe.

Για να υπολογίσουμε τις ελλιπείς τιμές του “smoking status”, ακολουθούμε παρόμοια μέθοδο, με μόνη διαφορά ότι χρησιμοποιούμε το μοντέλο ***LogisticRegression***, καθώς οι τιμές του γνωρίσματος αυτού είναι διακριτές, που αντιστοιχούν σε αλφαριθμητικά.

Στο τερματικό εμφανίζονται το πλήθος των NaN τιμών για τα δύο γνωρίσματα πριν και μετά την αξιοποίηση των μοντέλων για τον υπολογισμό των τιμών αυτών. Επίσης, δίνεται η δυνατότητα να τυπώσουμε τις προβλεπόμενες τιμές για το “bmi”.

**Αποτελέσματα:**



Εικόνα β: Πλήθος των NaN τιμών του smoking\_status πριν και μετά

Εικόνα a: Προβλεπόμενα αποτελέσματα bmi και το πλήθος των NaN τιμών πριν και μετά

#### Ερώτημα Β4

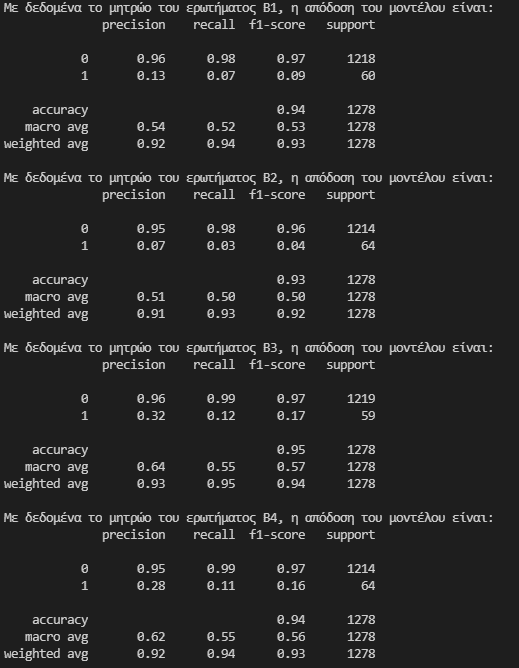
Δημιουργούμε τη συνάρτηση ***exerciseB\_4***. Η διαδικασία που ακολουθούμε σε αυτό το ερώτημα είναι παρόμοιας λογικής με αυτή του ερωτήματος 3 με τις εξής διαφορές:

* Για τον υπολογισμό των ελλιπών τιμών του “bmi” χρησιμοποιούμε το μοντέλο ***KNeighborsRegressor***, ενώ για τις ελλιπείς τιμές του “smoking status” χρησιμοποιούμε το μοντέλο ***KNeighborsClassifier***.
* Αφαιρούμε αρχικά από το dataframe τις στήλες που περιέχουν αλφαριθμητικές τιμές.

### 2.1.4 Ερώτημα Γ

Δημιουργούμε τη συνάρτηση ***exerciseC***, όπου θα αξιοποιήσουμε τα dataframes, τα οποία προέκυψαν από την εκτέλεση των συναρτήσεων του ερωτήματος Β με σκοπό να κατηγοριοποιήσουμε τα παραδείγματα μας και να συγκρίνουμε την ακρίβεια του ταξινομητή ανάλογα με το dataframe που του παρέχουμε. Για την κατηγοριοποίηση θα χρησιμοποιήσουμε το μοντέλο ***RandomForestClassifier***.

Για κάθε dataframe των ερωτημάτων Β1, Β2, Β3 και Β4, το διαχωρίζουμε σε 2 dataframes, όπου το πρώτο περιέχει όλα τα παραδείγματα χωρίς τις στήλες “stroke” και “id”, ενώ το δεύτερο περιέχει τη στήλη stroke. Στη συνέχεια, διαχωρίζουμε τα δεδομένα σε training και test data με αναλογία 75%-25%. Εκπαιδεύουμε το μοντέλο και προβλέπουμε την κατηγορία στην οποία ανήκει κάθε παράδειγμα και εκτυπώνουμε στο τερματικό τα αποτελέσματα των μετρικών καλώντας την συνάρτηση ***classification\_report***.



Εικόνα : Αποτελέσματα μετρικών ανά κατηγορία και συνολικά

Παρατηρούμε τα εξής:

* Το μοντέλο μας είναι ιδιαίτερο αποδοτικό όσον αφορά την κατηγοριοποίηση των παραδειγμάτων, τα οποία ανήκουν στην κατηγορία 0 σε όλες τις περιπτώσεις
* Αντιθέτως, κατά την κατηγοριοποίηση των παραδειγμάτων που ανήκουν στην κατηγορία 1, τα αποτελέσματα των μετρικών είναι αρκετά χαμηλά. Αυτό οφείλεται, κατά πάσα πιθανότητα, στο γεγονός ότι το πλήθος των παραδειγμάτων της συγκεκριμένης κατηγορίας είναι υπερβολικά μικρό σε σύγκριση με το πλήθος των παραδειγμάτων της κατηγορίας 1.

## 2.2 Άσκηση 2