Εξόρυξη Δεδομένων

Υλοποιητική Εργασία

Ον/μο: Σιάννας Σπυρίδων

Α.Μ.: 1053718

1. Εισαγωγή

Για την υλοποίηση αυτής της εργασίας χρησιμοποιήθηκε η γλώσσα προγραμματισμού Python. Πιο συγκεκριμένα, χρησιμοποιήθηκε το περιβάλλον προγραμματισμού Visual Studio Code και έγινε χρήση διάφορων βιβλιοθηκών. Επίσης έγινε χρήση του git για version control και του GitHub για την εναπόθεση του κώδικα.

GitHub Repository: https://github.com/SpyrosSiannas/DataMIningProject-2021

Για την υλοποίηση του πρώτου ερωτήματος, έγινε χρήση του jupyter notebook καθώς παρέχει εύκολη πρόσβαση στην εμφάνιση των δεδομένων μας και δίνει τη δυνατότητα εξαγωγής όλων των γραφικών παραστάσεων και σημειώσεων σε ένα ευανάγνωστο .html αρχείο.

Για την ανάλυση των δεδομένων χρησιμοποιήθηκε κατά κόρον η βιβλιοθήκη pandas καθώς και οι scikit-learn και keras/tensorflow.

Προτείνεται το κατέβασμα του project από το github (περιέχει τα pre-trained models)

1.1 Οδηγίες εγκατάστασης

Για την εγκατάσταση του project είναι απαραίτητη η γλώσσα προγραμματισμού python. Αφού βεβαιώσετε ότι υπάρχει εγκατεστημένη στο σύστημά σας, ακολουθήστε τα κάτωθι βήματα:

Θεωρώντας ότι βρισκόμαστε στο βασικό φάκελο του repository

- Κατασκευή ενός virtual environment:

```
python -m venv env
```

- Εγκατάσταση των απαιτούμενων βιβλιοθηκών από το αρχείο *requirements.txt:*

```
pip install -r requirements.txt
```

- Ενεργοποίηση του venv μέσω του command line:

```
env/Scripts/activate.bat (Windows)

env/bin/activate (MacOS)

. env/bin/activate (Linux)
```

Αφού ολοκληρωθεί η διαδικασία, η εργασία είναι έτοιμη. Εάν τρέξετε το project από κάποιο IDE, βεβαιωθείτε ότι χρησιμοποιεί την Python του Virtual Environment.

1.2 Περίληψη της δομής του παραδοτέου

Στο παραδοτέο περιέχονται τα εξής αρχεία.

- Ένα jupyter notebook με το πρώτο ερώτημα της άσκησης

- Η html εκδοχή του Notebook για ευκολότερη ανάγνωση
- Το αρχείο main.py και ο φάκελος src όπου περιέχεται ο κώδικας που υλοποιεί το δεύτερο ερώτημα της άσκησης
- Ο φάκελος dataset με τα δεδομένα της εργασίας
- Τα έγγραφα των εκφωνήσεων

2. Ερώτημα 1

Όλες οι γραφικές παραστάσεις, μαζί με σύντομες περιγραφές των μπορούν να βρεθούν στο αρχείο *main-notebook.html*

Α) Το πρώτο υπο-ερώτημα αφορά την ανάλυση και την προ-επεξεργασία ενός συνόλου δεδομένων. Αρχικά υλοποιήθηκε μία βοηθητική συνάρτηση για τον εύκολο σχεδιασμό των γραφικών παραστάσεων των συχνοτήτων των δεδομένων μας.

Έπειτα, αφαιρέθηκε η στήλη *id* καθώς δε φέρει κάποια σημασία στο σύνολο των δεδομένων μας και δε μπορεί να μεταφραστεί σε γνώση.

Εν συνεχεία, υπολογίστηκε το μητρώο συσχετίσεων και πραγματοποιήθηκε γραφική αναπαράσταση των συχνοτήτων, καθώς και αναπαράσταση τιμών στις οποίες η συσχέτιση είναι νοηματικά ορθή και επιβεβαιώνεται από τον πίνακα συσχετίσεων.

Β) Αφού είδαμε γραφικά τα δεδομένα μας και τα αναλύσαμε, πρέπει να τα ετοιμάσουμε για την εξαγωγή γνώσης με μία προ-επεξεργασία. Σε μία αρχική φάση, πρέπει να διαχειριστούμε τις τιμές που απουσιάζουν. Οι στήλες στις οποίες απουσιάζουν τιμές είναι ο δείκτης μάζας σώματος (**bmi)** καθώς και η τιμή **Unknown** στη στήλη που περιγράφει τις καπνιστικές συνήθειες των υποκειμένων. Μία αρχική προσέγγιση είναι να αφαιρεθούν τελείως οι στήλες αυτές, πράγμα που η βιβλιοθήκη pandas καθιστά πολύ απλό. Αυτή η προσέγγιση έχει όμως σαν αποτέλεσμα απώλεια σημαντικής γνώσης, οπότε μεταβαίνουμε σε μία άλλη λύση.

Μία άλλη προσέγγιση είναι η συμπλήρωση των κενών αυτών τιμών βάσει της μέσης τιμής της στήλης. Αυτό μας δίνει μία καλύτερη προσέγγιση από την πρώτη λύση αλλά ακόμη δεν είναι ικανοποιητική.

Έπειτα, μπορούμε να χρησιμοποιήσουμε γνωστούς αλγορίθμους για να κάνουμε Impute τις συγκεκριμένες τιμές. Ένας τέτοιος αλγόριθμος είναι η Γραμμική Παλινδρόμηση. Πρόκειται για έναν αλγόριθμο ο οποίος προσπαθεί να προσαρμοστεί στα δεδομένα μας και να δώσει μία γραμμική συσχέτιση μεταξύ δύο τιμών.

$$Y = \alpha X$$

Για εξαρτημένη μεταβλητή, επιλέχθηκε η ηλικία έχουσα τη μεγαλύτερη τιμή στον πίνακα συσχέτισης.

Ένας άλλος αλγόριθμος που μπορεί να φανεί χρήσιμος είναι ο K-Nearest-Neighbors. Εδώ αρχικά εφαρμόσαμε μία κωδικοποίηση των κατηγορικών δεδομένων μας ώστε να είναι καθαρά αριθμητικά. Θα μπορούσε να εφαρμοστεί και One Hot Encoding, όμως επιλέχθηκε η Ordinal κωδικοποίηση. Αφού έγινε αυτή η διαδικασία, εφαρμόστηκε KNN imputation με K=5.

Από αυτόν τον αλγόριθμο εξήγαμε δύο σύνολα δεδομένων, ένα έχον τη στήλη bmi γεμισμένη από το αποτέλεσμα της γραμμικής παρεμβολής και ένα στο οποίο τη διώξαμε.

Γ) Ο αλγόριθμος RandomForest της sklearn παρουσίασε πολλά προβλήματα καθώς οι κλάσεις μας δεν είναι καθόλου ισορροπημένες με αποτέλεσμα κάποια δέντρα του κατηγοριοποιητή πολλές φορές να μη βλέπουν αντικείμενα της κλάσης 1 (π.χ.) και να οδηγούμαστε σε false negatives. Μερικές βελτιώσεις είναι η απενεργοποίηση του bootstrap και ο ορισμός των max_features σε 1 όμως και πάλι δεν είχαμε καλό αποτέλεσμα. Μία τελευταία λύση που θα μπορούσε να υλοποιηθεί σαν μελλοντική προσθήκη είναι μία δική μου υλοποίηση της κλάσης RandomForest όπου θα εφαρμόζεται μία κανονικοποίηση στα δεδομένα εισόδου για να έχουμε μία καλή ισορροπία στα δεδομένα μας και σωστό σχηματισμό δάσους.

Όλες οι μετρικές για τον αλγόριθμο προκύπτουν οριακά μηδενικές.

3. Ερώτημα 2

Το δεύτερο ερώτημα έχει να κάνει με την επεξεργασία φυσικής γλώσσας και την κατηγοριοποίηση κειμένων ως επιθυμητών ή μη. Υλοποιήθηκε μία κλάση νευρωνικού δικτύου, οι βασικές ρυθμίσεις της οποίας βρίσκονται στο αρχείο *cfg.py*.

Για την υλοποίηση αυτού του ερωτήματος, εκπαιδεύτηκε ένα μοντέλο Word2Vec. Μία προσθήκη που θα μπορούσε να γίνει στο μοντέλο αυτό θα ήταν ένας lemmatizer ώστε οι λέξεις με ίδια γραμματική ρίζα να θεωρούνται ίδιες από το μοντέλο και να μη λαμβάνουν διαφορετική κωδικοποίηση.

Μετά την κωδικοποίηση και το διαχωρισμό των δεδομένων μας σε 0.75 για εκπαίδευση και 0.25 για επαλήθευση, περνάμε το σύνολο των δεδομένων εκπαίδευσης από ένα νευρωνικό δίκτυο με ένα επίπεδο εισόδου με συνάρτηση ενεργοποίησης ReLU, δύο κρυφά επίπεδα με την ίδια συνάρτηση ενεργοποίησης και ένα επίπεδο εξόδου μήκους ένα με σιγμοειδή συνάρτηση ενεργοποίησης.

Η συνάρτηση κόστους που επιλέχθηκε είναι η binary cross-entropy καθώς η κατηγοριοποίηση γίνεται μεταξύ των τιμών 0 και 1.

Το δίκτυο εκπαιδεύεται με μία ακρίβεια εκπαίδευσης ~95%, ακρίβεια δοκιμών ~65% και μετρικές

$$Precision = 0.4273255813953488$$

$$Recall = 1.0$$

$$F1 = 0.5987780040733197$$

Ένας lemmatizer πιθανότατα θα αύξανε την τελική απόδοση όπως αναφέρθηκε και ανωτέρω.