ΕΡΓΑΣΤΗΡΙΑΚΗ ΆΣΚΗΣΗ

Εξόρυξη Δεδομένων και Αλγόριθμοι Μάθησης

Βαζαίος Στυλιανός (1054284)

Σταυρόπουλος Παναγιώτης (1058085)

Περιεχόμενα

Εισαγωγικά:

Περιβάλλον υλοποίησης: PyCharm 2020.1.2

Interpreter: Python 3.7

Χρησιμοποιούμενες βιβλιοθήκες:

- Keras
- Keras-Preprocessing
- Markdown
- PyYAML
- Werkzeug
- absl-py
- astunparse
- cachetools
- certifi
- chardet
- click
- cycler
- gast
- google-auth
- google-auth-oauthlib
- google-pasta
- grpcio
- h5py
- idna
- importlib-metadata
- joblib
- kiwisolver
- matplotlib
- nltk
- numpy
- oauthlib
- opt-einsum
- pandas
- pip
- protobuf
- pyasn1
- pyasn1-modules
- pyparsing
- python-dateutil
- pytz
- regex
- requests
- requests-oauthlib

- rsa
- scikit-learn
- scipy
- setuptools
- six
- tensorboard
- tensorboard-plugin-wit
- tensorflow
- tensorflow-estimator
- termcolor
- tqdm
- urllib3
- wheel
- wrapt
- zipp

Βήματα εγκατάστασης:

1) Κατέβασμα και εγκατάσταση της Python 3.7 (https://www.python.org/downloads/).
2) Κατέβασμα και εγκατάσταση του PyCharm 2020.1.2 (https://www.jetbrains.com/pycharm/download/#section=windows).
3) Εγκατάσταση βιβλιοθηκών από το System (Ctrl + Alt + S) ή το terminal του Pycharm.

Οι κώδικες των αρχείων data_mining.py και erot2.py είναι πλήρη σχολιασμένοι για να είναι εύκολο ο χρήστης να καταλαβαίνει τι κάνει η κάθε γραμμή κώδικα. Στο data_mining.py επιπλέον εμφανίζεται στο terminal όλη η διαδικασία που ακολουθείται από το ερώτημα Α μέχρι το ερώτημα Β εμφανίζοντας τα αντίστοιχα dataframes, λίστες δεδομένων, αποτελέσματα μετρήσεων, και μεταβλητές που χρησιμοποιούμε. Στο τέλος της εκτέλεση του data_mining.py εμφανίζεται στο terminal τα αποτελέσματα τ ποιότητας κατηγοριοποίησης με τις νέες μετρικές του ερωτήματος Β.

Η ανάλυση του κώδικα στο erot2.py σε συναρτήσεις και υποφακέλους φάνηκε περιττός, διότι πρόκειται για πρόγραμμα λιγότερο από 100 γραμμές, οι απαραίτητες συναρτήσεις βρίσκονται ήδη στις βιβλιοθήκες και μια τέτοια ανάλυση θα οδηγούσε σε χαμηλής ποιότητας κώδικα, κακογραμμένο και ασαφή.

Ερώτημα 1

Μέρος Α

Η άσκηση αποτελεί υλοποίηση «supervised learning» με βάση τα ζητούμενα του Α ερωτήματος χωρίσαμε το dataset σε training-test σε αναλογία 75%-25% . Παράλληλα προσπαθήσαμε να βρούμε το καλύτερο δυνατό μοντέλο SVC πειραματιζόμενοι με τις παραμέτρους εισόδου των αλγορίθμων κατηγοριοποίησης SVM. Με βάση τις δοκιμές που εκτελέσαμε επιλέξαμε τις πιο ιδανικές παραμέτρους με βάση τα αποτελέσματα από τις μετρικές f1 score , precision και recall τα οποία είναι τα εξής:

kernel	poly
degree	5
gamma	scale
class_weight	None
coef0	5.2
cache_size	500
accuracy(scrikit)	0.6375
f1 score	0.6118
precision	0.5917
recall	0.6375

Ακολουθούν ορισμένα από τα πειράματα που κάναμε για να βρούμε τις πιο ιδανικές παραμέτρους.

kernel	linear	linear	poly	poly	poly	poly	poly	poly	poly	rbf	poly	poly	poly
degree	default	default	default	default	default	default	default	default	4	4	5	5	5
gamma	default	default	default	default	default	default	default	scale	scale	scale	scale	scale	scale
class_weight	balanced	None	None	balanced	balanced	balanced	balanced	balanced	balanced	balanced	balanced	None	None
coef0	default	default	default	default	default	2	4	4	5.2	5.2	5.2	5.2	6
cache_size	default	default	default	default	500	500	500	500	500	500	500	500	500
accuracy(scrikit)	0.4150	0.635	0.5025	0.1625	0.1625	0.32	0.3575	0.3575	0.4225	0.2775	0.4925	0.6375	0.64
f1 score	0.4678	0.5948	0.4347	0.2301	0.2301	0.3912	0.4201	0.4201	0.4795	0.3445	0.5226	0.6118	0.6132
precision	0.6000	0.5733	0.4819	0.5388	0.5388	0.5653	0.5695	0.5695	0.5864	0.4683	0.6175	0.5917	0.5938
recall	0.4150	0.635	0.5025	0.1625	0.1625	0.32	0.3575	0.3575	0.4225	0.2775	0.4925	0.6375	0.64

Μέρος Β

Για τα υπό-ερωτήματα 1,2,3,4 του ερωτήματος Β χρησιμοποιήθηκαν dataframes και λίστες για να εκπληρωθούν οι απαιτήσεις τους. Στο αρχείο data_mining.py το ερώτημα β1 αντιστοιχεί στην συνάρτηση b1(), το ερώτημα β2 στην b2, το ερώτημα β3 στην b3 και τέλος το ερώτημα β4 στην συνάρτηση b4() . Οι συναρτήσεις b2,b3 και b4 παίρνουν σαν όρισμα την λίστα που προκύπτει από την μετατροπή του dataframe που αφαιρέσαμε 33% των τιμών τις στήλης pH σε πίνακα, ενώ η b1 το X_train dataframe. Οι τιμές που διαγράφτηκαν αντικαταστάθηκαν με το string ‘zero’ . Κανονικά θα μπορούσαμε να χρησιμοποιούσαμε το None της python αλλά για να είναι εύκολη η επαλήθευση της λειτουργίας του προγράμματος χρησιμοποιήσαμε το ‘zero’ για να βλέπουμε τα αποτελέσματα του terminal πιο εύκολα. Επιπλέον πρέπει να τονίσουμε πως για το ερώτημα β3 στο logistic regression χρησιμοποιήσαμε max_iter=1000000 λόγω του μεγέθους του αρχείου και των πράξεων που έπρεπε να γίνουν καθώς για να έχουμε πιο σωστά δεδομένα ως αποτέλεσμα προεπεξεργαστήκαμε τα δεδομένα που είχαμε . Δηλαδή κάναμε πολλαπλασιασμούς για να τα δεκαδικά ψηφία των float τιμών να περαστούν σαν integers για να έχουμε μεγαλύτερη

ακρίβεια. Στην συνέχει στο τέλος των αποτελεσμάτων που θέλουμε τα επαναφέρουμε στην αρχική τους κατάσταση και τα προσθέτουμε στο X_train του β3.

Στα νέα μητρώα που προέκυψαν με βάση το SVC του Ερωτήματος Α για τις καλύτερες παραμέτρους του εκπαιδεύουμε πάλι το classifier και ελέγχουμε τα αποτελέσματα των μετρικών που έχουμε. Στην περίπτωση του β1 ερωτήματος χρειάστηκε να διαγράψουμε και την στήλη του pH του X_test του β1 αφού λείπει στο X_train του. Στα υπόλοιπα παρέμειναν ως έχει με την στήλη του pH. Τα αποτελέσματα των μετρικών είναι τα εξής:

Ερώτημα β1)

- Accuracy of b1: 0.64
- f1 score_B1: 0.6155590310590311
- Precision_B1: 0.5966738563281154
- Recall_B1: 0.64

Ερώτημα β2)

- Accuracy of b2: 0.635
- f1 score_B2: 0.6109335357952704
- Precision_B2: 0.5913589664307688
- Recall_B2: 0.635

Ερώτημα β3)

- Accuracy of b3: 0.6425
- f1 score_B3: 0.6154330379155671
- Precision_B3: 0.594815664580211
- Recall_B3: 0.6425

Με βάση αυτά τα δεδομένα που παρατηρούμε ,συμπεραίνουμε πως τα αποτελέσματα παραμένουν σχεδόν αμετάβλητα και διατηρούν ίδιες περίπου ποιότητες κατηγοριοποίησης.

Για το ερώτημα β4, δηλαδή η συνάρτηση b4(), έχει αρχίσει η ανάπτυξή του αλλά δεν έχει ολοκληρωθεί πλήρως. Για αυτό το λόγο οι μετρικές για την ποιότητα των αποτελεσμάτων του έχουν παραμείνει σαν σχόλια στον κώδικα στις γραμμές 465-469.

Ερώτημα 2

Υλοποίηση:

1) Διαβάζεται το ζητούμενο έγγραφο και εξάγονται οι πληροφορίες του.
2) Αφαιρούνται οι καταλήξεις των λέξεων, κρατώντας μόνο το θέμα τους (stemming) και αφαιρούνται οι αρκετά κοινές λέξεις που δεν προσφέρουν πληροφορία (stopwords removal).
3) Ανατίθεται το βάρος tf-idf.

4) Το νευρωνικό δίκτυο εκπαιδεύεται χρησιμοποιώντας το 75% του dataset. Επιλέγεται το Sequential, διότι έχουμε μοναδική είσοδο και έξοδο και προσφέρει μεγαλύτερη ακρίβεια, με αντάλλαγμα όμως μεγάλη υπολογιστική ισχύ.
5) Με το εναπομένον 25% μετριέται η απόδοσή του από τις μετρικές f1 score, precision και recall.
6) Εκτυπώνονται τα αποτελέσματα.

Τελικά αποτελέσματα:

Τα παρακάτω αποτελέσματα δεν αποτελούν τον μέσο όρο, διότι η εκτέλεση του αρχείου έχει χρόνο πάνω από 2,5 λεπτά. Πρόκειται όμως για έμπιστα αποτελέσματα, λόγω του μεγάλου όγκου δεδομένων στην είσοδο.

- loss: 0.1115
- accuracy: 0.9581
- F1 Score: 0.8915
- Precision Score: 0.8915
- Recall Score: 0.8915

- Runtime: 172.0215060710907 seconds

Τα τελικά αποτελέσματα δείχνουν αρκετά καλές επιδώσεις (σχεδόν 90% επιτυχία) οπότε καταλαβαίνουμε ότι το νευρωνικό δίκτυο λειτουργεί σωστά.