Τμήμα Μηχανικών Η/Υ & Πληροφορικής
Πανεπιστήμιο Πατρών

Εξόρυξη Δεδομένων και Αλγόριθμοι Μάθησης

Εργαστηριακή Άσκηση
Εαρινό Εξάμηνο 2020-21

Δασούλας Ιωάννης – 1053711 – 5ο Έτος

Τμήμα Ηλεκτρολόγων Μηχανικών και τεχνολογίας Υπολογιστών

Εισαγωγή

Τα ερωτήματα της άσκησης έγιναν σε γλώσσα Python 3.7, χρησιμοποιώντας και συγκεκριμένες βιβλιοθήκες. Σε κάθε ερώτημα εξηγούνται ξεχωριστά οι βιβλιοθήκες που χρησιμοποιήθηκαν. Για την ορθή λειτουργία των προγραμμάτων εξηγούνται ξεχωριστά οι βιβλιοθήκες που χρησιμοποιήθηκαν. Για την ορθή λειτουργία των προγραμμάτων, χρειάζεται τα αντίστοιχα csv αρχεία να βρίσκονται στον ίδιο φάκελο με τα προγράμματα.

Ερώτημα 1

Α.

Για το πρώτο ερώτημα χρησιμοποιήθηκε η βιβλιοθήκη pandas για ανάγνωση και αποθήκευση των δεδομένων από το csv αρχείο (healthcare-dataset-stroke-data.csv), αλλά και για την δημιουργία των γραφημάτων, η βιβλιοθήκη matplotlib για την ταυτόχρονη τύπωση των γραφημάτων και η βιβλιοθήκη sys για την έξοδο από το πρόγραμμα σε περίπτωση απουσίας του csv αρχείου.

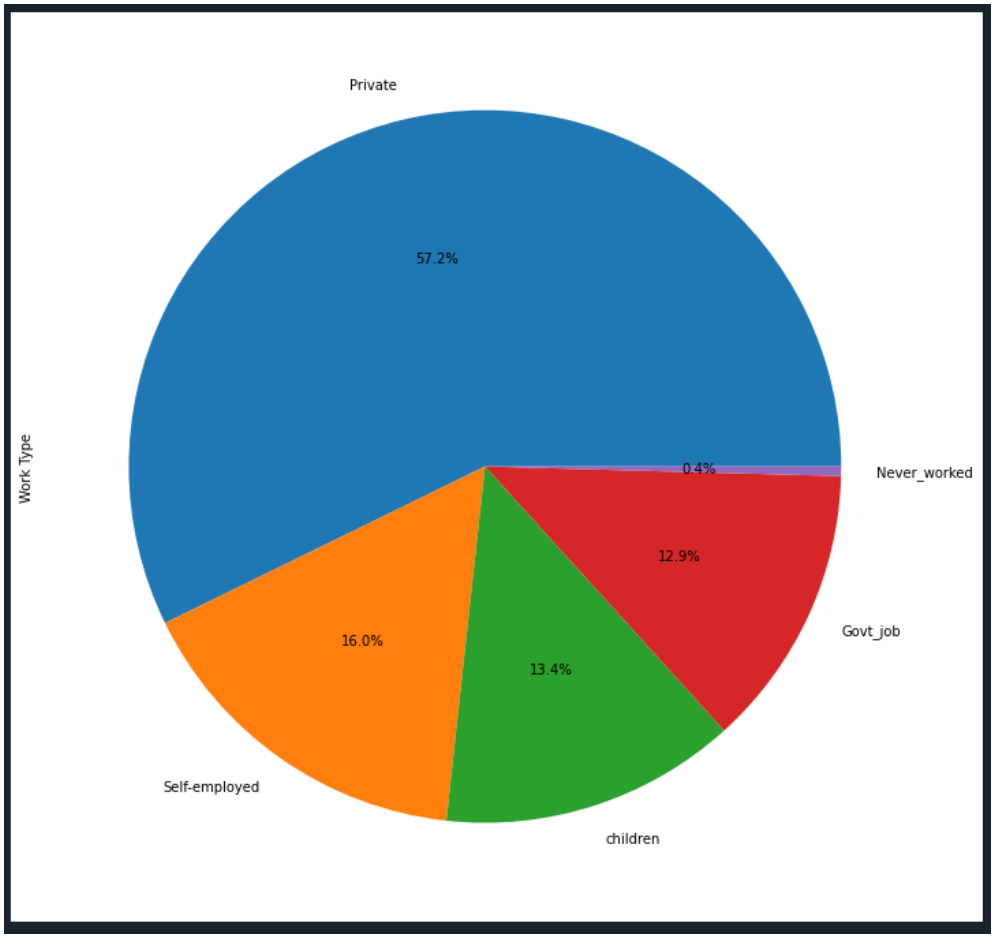
Απαραίτητες εντολές εγκατάστασης πριν την εκτέλεση:

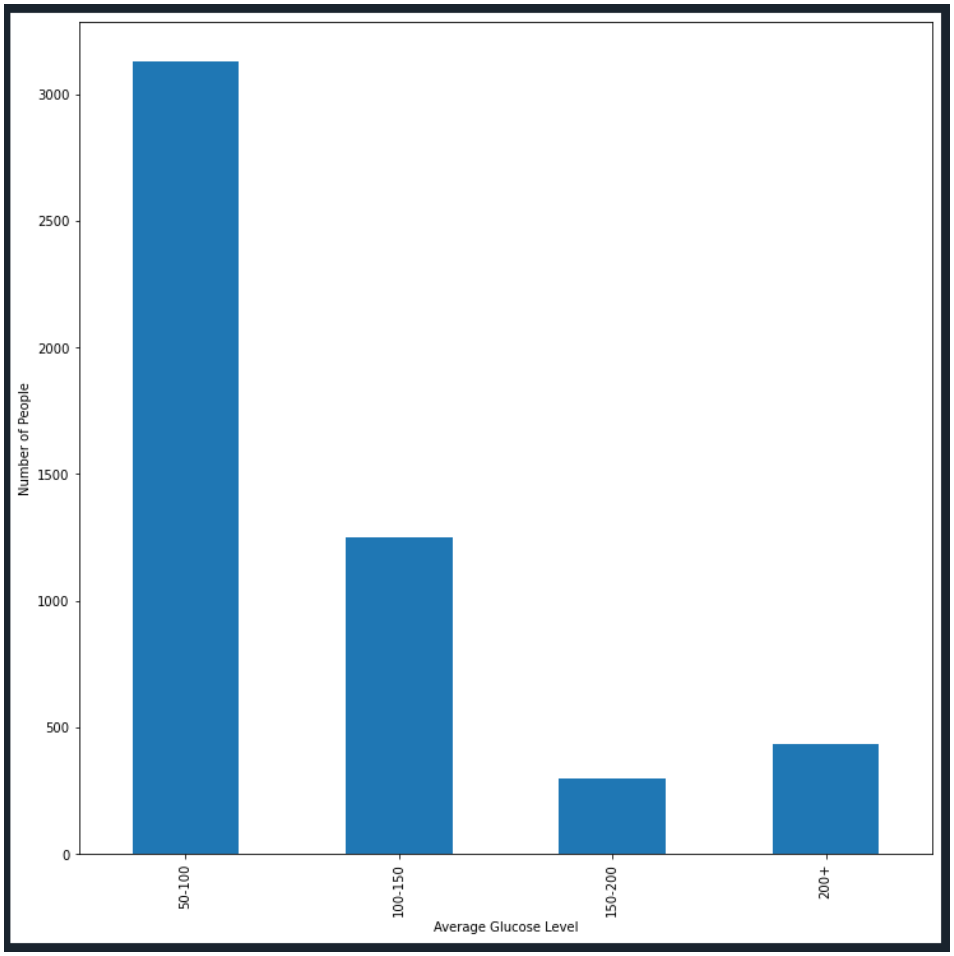
- pip install pandas
- pip install matplotlib

Αρχικά φορτώνεται το dataset και αποθηκεύεται σε ένα pandas dataframe, τυπώνονται οι τύποι των δεδομένων, το σχήμα του dataset και γίνεται αναζήτηση για ίδιες σειρές δεδομένων αλλά και για στήλες που περιέχουν ‘None’ τιμές. Έπειτα δημιουργούνται μερικά γραφήματα για κάθε στήλη που φανερώνουν τα πεδία των τιμών και την συχνότητα τους στην στήλη.

Παρατηρήθηκε ότι δεν υπάρχουν ίδιες γραμμές στο dataset και ότι μόνο μία στήλη περιέχει ‘None’ τιμές, η στήλη ‘bmi’.

Παραδείγματα γραφημάτων που εμφανίζονται:





B.

Για το δεύτερο ερώτημα χρησιμοποιήθηκε η βιβλιοθήκη pandas για ανάγνωση και αποθήκευση των δεδομένων από το csv αρχείο (healthcare-dataset-stroke-data.csv), η βιβλιοθήκη sklearn για την δημιουργία του 'K-Nearest Neighbors' μοντέλου, η βιβλιοθήκη numpy για την ανίχνευση των 'None' τιμών και η βιβλιοθήκη sys για την έξοδο από το πρόγραμμα σε περίπτωση απουσίας του csv αρχείου.

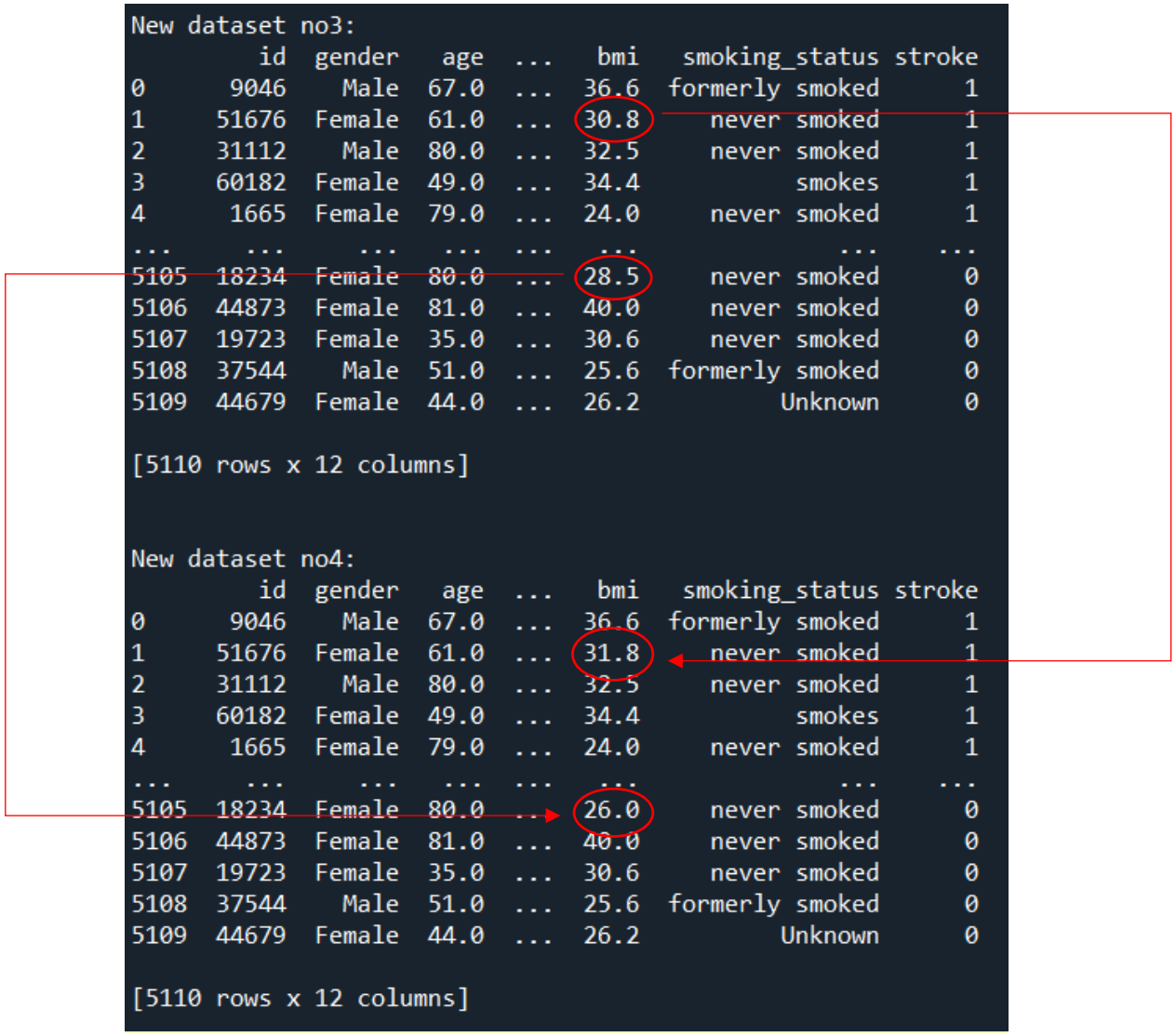
Απαραίτητες εντολές εγκατάστασης πριν την εκτέλεση:

- pip install pandas
- pip install sklearn

Αρχικά φορτώνεται το dataset και αποθηκεύεται σε ένα pandas dataframe, και δημιουργούνται 4 διαφορετικές μέθοδοι-συναρτήσεις που διαχειρίζονται με ξεχωριστό τρόπο η κάθε μία τις τιμές που λείπουν στην στήλη 'bmi' δημιουργώντας έτσι 4 νέα dataframes, δηλαδή 4 νέα datasets. Επίσης, υπάρχει και η τιμή 'unknown' σε κάποια παραδείγματα του 'skoking status' αλλά επιλέχθηκε να μην επεξεργαστεί μιας και το ερώτημα αναφέρεται σε αριθμητική επεξεργασία της στήλης που περιέχει 'None' τιμές, κάτι που δεν θα μπορούσε να γίνει στην στήλη 'smoking status' που περιέχει αλφαριθμητικές τιμές. Εναλλακτικά, θα μπορούσε να γίνει απλά διαγραφή των παραδειγμάτων με την τιμή 'Unknown' στην στήλη 'smoking status'.

Το πρώτο νέο dataset δημιουργείται απλά αφαιρώντας τις στήλες που δεν υπάρχει τιμή. Το δεύτερο, αντικαθιστώντας τις τιμές που λείπουν με τον μέσο όρο της στήλης. Στην τρίτη μέθοδο γίνεται η υπόθεση ότι το bmi εξαρτάται από τα επίπεδα γλυκόζης ('avg_glucose_level'), όπως βρέθηκε σε σχετική έρευνα. Εναλλακτικά, μπορεί να θεωρηθεί ότι εξαρτάται από την ηλικία ή να βρεθούν οι συσχετίσεις του 'bmi' με τα υπόλοιπα δεδομένα και να βρεθεί με ποιο ή ποια σχετίζεται περισσότερο. Ορίζοντας ότι η στήλη 'bmi' παίρνει τιμές στον άξονα x και η στήλη 'avg_glucose_level' παίρνει τιμές στον άξονα y, εφαρμόζεται linear regression σύμφωνα με τον τύπο: y = x*slope + bias. Αφού βρίσκονται οι τιμές slope και bias, μέσω των σχετικών τύπων τις μεθόδου και των υπόλοιπων δεδομένων, υπολογίζονται όλες οι τιμές που λείπουν, βάσει του παραπάνω τύπου, και δημιουργείται, έτσι το τρίτο dataset. Στην τέταρτη μέθοδο, γίνεται ξανά η υπόθεση ότι το bmi εξαρτάται από τα επίπεδα γλυκόζης ('avg_glucose_level'), εκπαιδεύεται το μοντέλο KNeighborsRegressor με τις παρούσες τιμές που φορτώνεται από την βιβλιοθήκη sklearn και στην συνέχεια γίνεται η πρόβλεψη των τιμών που λείπουν μέσω του μοντέλου, δημιουργώντας έτσι το τέταρτο dataset.

Συγκρίνοντας τα 3 τελευταία datasets που δημιουργήθηκαν, τα οποία αφορούν πρόβλεψη των τιμών που λείπουν, παρατηρείται ότι είναι κοντά αριθμητικά οι τιμές που υπολογίζονται μεταξύ των μεθόδων για ένα ίδιο παράδειγμα του dataset και είναι εντός του πεδίου τιμών της στήλης, δηλαδή και οι 3 μέθοδοι κάνουν καλή δουλειά στην εύρεση νέων τιμών. Στην παρακάτω εικόνα φαίνονται 2 από τα 4 νέα datasets (αυτά των 2 τελευταίων μεθόδων) και είναι κυκλωμένες οι τιμές που έλειπαν και υπολογίστηκαν, όπου και φαίνεται πως είναι κοντά αριθμητικά.


New dataset no3:
id gender age ... bmi smoking_status stroke
0 9046 Male 67.0 ... 36.6 formerly smoked 1
1 51676 Female 61.0 ... 30.8 never smoked 1
2 31112 Male 80.0 ... 32.5 never smoked 1
3 60182 Female 49.0 ... 34.4 smokes 1
4 1665 Female 79.0 ... 24.0 never smoked 1
... ...
5105 18234 Female 80.0 ... 28.5 never smoked 0
5106 44873 Female 81.0 ... 40.0 never smoked 0
5107 19723 Female 35.0 ... 30.6 never smoked 0
5108 37544 Male 51.0 ... 25.6 formerly smoked 0
5109 44679 Female 44.0 ... 26.2 Unknown 0
[5110 rows x 12 columns]
New dataset no4:
id gender age ... bmi smoking_status stroke
0 9046 Male 67.0 ... 36.6 formerly smoked 1
1 51676 Female 61.0 ... 31.8 never smoked 1
2 31112 Male 80.0 ... 32.5 never smoked 1
3 60182 Female 49.0 ... 34.4 smokes 1
4 1665 Female 79.0 ... 24.0 never smoked 1
... ...
5105 18234 Female 80.0 ... 26.0 never smoked 0
5106 44873 Female 81.0 ... 40.0 never smoked 0
5107 19723 Female 35.0 ... 30.6 never smoked 0
5108 37544 Male 51.0 ... 25.6 formerly smoked 0
5109 44679 Female 44.0 ... 26.2 Unknown 0
[5110 rows x 12 columns]

Γ.

Για το τρίτο ερώτημα χρησιμοποιήθηκε η βιβλιοθήκη pandas για ανάγνωση και αποθήκευση των δεδομένων από το csv αρχείο (healthcare-dataset-stroke-data.csv), η βιβλιοθήκη sklearn για την δημιουργία του 'K-Nearest Neighbors' μοντέλου, για τον διαχωρισμό του dataset σε δεδομένα εκπαίδευσης και δεδομένα πρόβλεψης, για την χρήση του μοντέλου RandomForestClassifier, για την χρήση του LabelEncoder που βοηθάει στην μετατροπή των αλφαριθμητικών δεδομένων σε αριθμητικά και για την χρήση του classification_report που εφαρμόζει τις ζητούμενες μετρικές στα αποτελέσματα. Επίσης, φορτώνεται η βιβλιοθήκη numpy για την ανίχνευση των 'None' τιμών και η βιβλιοθήκη sys για την έξοδο από το πρόγραμμα σε περίπτωση απουσίας του csv αρχείου.

Απαραίτητες εντολές εγκατάστασης πριν την εκτέλεση:

- pip install pandas
- pip install sklearn

Στο τρίτο ερώτημα γίνεται η πρόβλεψη για το αν ένας ασθενής είναι επιρρεπής ή όχι να πάθει εγκεφαλικό χρησιμοποιώντας Random Forest. Αρχικά δημιουργούνται πάλι τα 4 νέα datasets με τις 4 μεθόδους. Έπειτα, για κάθε dataset, μετατρέπονται τα αλφαριθμητικά δεδομένα σε αριθμητικά ώστε να μπορεί να γίνει η εκπαίδευση, μέσω του LabelEncoder() της sklearn. Κάθε ξεχωριστή αλφαριθμητική τιμή αντικαθίσταται από έναν ακέραιο αριθμό, ξεκινώντας από το 0. Αφού, πλέον το dataset αποτελείται μόνο από αριθμητικά δεδομένα, χωρίζεται σε σύνολο εκπαίδευσης (75%) και σύνολο πρόβλεψης (25%) μέσω της train_test_split() της sklearn. Έπειτα, φορτώνεται το μοντέλο RandomForestClassifier, επίσης από την sklearn, και εκπαιδεύεται με τα δεδομένα εκπαίδευσης. Στην συνέχεια, γίνεται η πρόβλεψη για τα υπόλοιπα δεδομένα και τυπώνονται τα αποτελέσματα των μετρικών μέσω της classification_report() της sklearn. Τα αποτελέσματα είναι τα εξής:

Method 1 results:

	precision	recall	f1-score	support
0	0.96	0.99	0.98	1223
1	0.30	0.05	0.09	55
accuracy			0.95	1278
macro avg	0.63	0.52	0.53	1278
weighted avg	0.93	0.95	0.94	1278

Method 2 results:

	precision	recall	f1-score	support
0	0.95	1.00	0.97	1211
1	0.60	0.04	0.08	67
accuracy			0.95	1278
macro avg	0.77	0.52	0.53	1278
weighted avg	0.93	0.95	0.93	1278

Method 3 results:

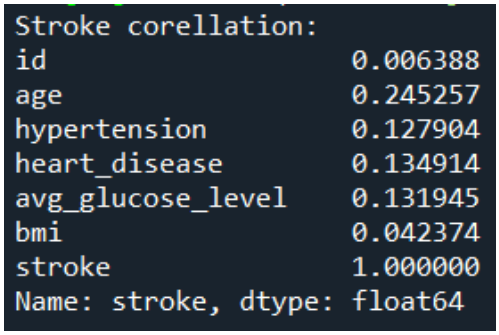
	precision	recall	f1-score	support
0	0.95	1.00	0.97	1213
1	0.00	0.00	0.00	65

accuracy			0.95	1278
macro avg	0.47	0.50	0.49	1278
weighted avg	0.90	0.95	0.92	1278

Method 4 results:

	precision	recall	f1-score	support
0	0.95	1.00	0.97	1210
1	0.17	0.01	0.03	68
accuracy			0.94	1278
macro avg	0.56	0.51	0.50	1278
weighted avg	0.91	0.94	0.92	1278

Παρατηρώντας τα αποτελέσματα, γίνεται προφανές ότι ενώ το μοντέλο προβλέπει με επιτυχία τις περιπτώσεις μη ύπαρξης stroke, δεν έχει επιτυχία στην πρόβλεψη για τις περιπτώσεις που θα έπρεπε να προβλέψει ότι υπάρχει stroke. Αλλάζοντας τις παραμέτρους του προβλήματος, είτε δοκιμάζοντας άλλα ποσοστά διαχωρισμού των δεδομένων είτε προσθέτοντας περισσότερα δέντρα απόφασης στη μέθοδο Random Forest, δεν παρατηρήθηκε ιδιαίτερη βελτίωση, παρά μόνο μικρή. Ένας λόγος είναι ο μικρός αριθμός παραδειγμάτων στο dataset με τιμή ‘1’ στο stroke με αποτέλεσμα να μην είναι επαρκή για την εκπαίδευση. Ένας άλλος λόγος είναι ότι η ύπαρξη stroke δεν έχει ισχυρό συσχετισμό με τα δεδομένα όπως μπορεί να υπολογιστεί. Σχετίζεται περισσότερο με την στήλη ‘age’ αλλά όχι σε μεγάλο ποσοστό. Έτσι, παρά την σωστή εκπαίδευση του μοντέλου, δεν μπορεί να γίνουν σωστές προβλέψεις για το stroke σε κανένα από τα datasets.



Ερώτημα 2

Για το δεύτερο πρόβλημα χρησιμοποιήθηκε η βιβλιοθήκη pandas για ανάγνωση και αποθήκευση των δεδομένων από το csv αρχείο (healthcare-dataset-stroke-data.csv), η βιβλιοθήκη sklearn για τον διαχωρισμό του dataset σε δεδομένα εκπαίδευσης και δεδομένα πρόβλεψης, για την χρήση του sklearn.metrics που εφαρμόζει τις ζητούμενες μετρικές στα αποτελέσματα και για την χρήση του StandardScaler() που κανονικοποιεί τα δεδομένα. Επίσης, φορτώνεται η βιβλιοθήκη numpy για την δημιουργία numpy arrays και η βιβλιοθήκη sys για την έξοδο από το πρόγραμμα σε περίπτωση απουσίας του csv αρχείου. Το μοντέλο Doc2Vec δημιουργείται με την βιβλιοθήκη gensim. Επίσης, χρησιμοποιείται η βιβλιοθήκη keras για την κατασκευή του νευρωνικού δικτύου που δέχεται τα διανύσματα των προτάσεων και προβλέπει αν είναι spam ή όχι. Τέλος, χρησιμοποιείται η ntlk για το tokenization των mails, τον διαχωρισμό τους, δηλαδή, σε λέξεις.

Απαραίτητες εντολές εγκατάστασης πριν την εκτέλεση:

- pip install pandas
- pip install sklearn
- pip install gensim==3.8.3
- pip install smart_open==2.0.0
- pip install keras
- pip install nltk

Αρχικά, φορτώνεται το αρχείο “spam_or_not_spam.csv” και αποθηκεύεται σε ένα pandas dataframe. . Στο περιεχόμενο των mail γίνεται tokenization, δηλαδή η πρόταση γίνεται λίστα λέξεων. Οι λίστες αυτές στην συνέχεια προστίθενται σε μία λίστα και τροφοδοτούν το μοντέλο Doc2Vec με την ομώνυμη συνάρτηση της gensim.

Το μοντέλο Doc2Vec επιλέχτηκε διότι η εργασία αφορά τη σχέση μεταξύ των mails και όχι μεταξύ λέξεων ενός συνόλου. Το Doc2Vec είναι ένα μοντέλο που αναπαριστά κάθε πρόταση ή σύνολο προτάσεων ως ένα διάνυσμα και αποτελεί την εξέλιξη του Word2Vec μοντέλου. Σε αντίθεση με το Word2Vec μοντέλο που παράγει ένα διάνυσμα για κάθε λέξη, το Doc2Vec παρέχει και τη δυνατότητα παραγωγής διανύσματος για κάθε πρόταση/παράγραφο, καθιστώντας το κατάλληλο για το συγκεκριμένο πρόβλημα. Τα διανύσματα παράγονται με νευρωνικό δίκτυο της βιβλιοθήκης και παρέχουν δείγμα της σχέσης μεταξύ των προτάσεων σε μορφή αριθμών, ανιχνεύοντας συνώνυμες λέξεις και προβλέποντας πιθανές πρόσθετες λέξεις για κάθε πρόταση. Τα διανύσματα επιλέγονται προσεκτικά, έτσι ώστε μια απλή μαθηματική συνάρτηση (η συνημιτονική ομοιότητα μεταξύ των διανυσμάτων) να δείχνει το επίπεδο εννοιολογικής ομοιότητας μεταξύ των προτάσεων που τα διανύσματα αναπαριστούν. Το μήκος του διανύσματος τίθεται ως παράμετρος στην συνάρτηση Doc2Vec, με μελέτες να έχουν δείξει πως όσο μεγαλύτερο, τόσο καλύτερο για την ακρίβεια των αποτελεσμάτων, κάτι που εξαρτάται και από το πλήθος των προτάσεων. Για τα 1505 mails του ερωτήματος, τέθηκε ίσο με 300.

Τα διανύσματα που προκύπτουν τροφοδοτούν νευρωνικό δίκτυο που δημιουργείται με την βιβλιοθήκη keras αφού χωριστούν σε σύνολο εκπαίδευσης και αξιολόγησης σύμφωνα με τις οδηγίες της άσκησης. Στο δίκτυο χρησιμοποιήθηκαν 4 dense layers, με 300 νευρώνες εισόδου, όσους και το μέγεθος των διανυσμάτων, και έναν νευρώνα εξόδου. Χρησιμοποιήθηκε η binary crossentropy συνάρτηση σφάλματος μιας και το πρόβλημα αφορά δυαδική κατηγοριοποίηση και η εκπαίδευση έγινε σε 100 εποχές. Το δίκτυο αξιολογήθηκε με τα παραδείγματα αξιολόγησης. Η μέτρηση της απόδοσης του δικτύου έγιναν με τις μετρικές της sklearn.metrics. Τα αποτελέσματα φανερώνουν τη σωστή λειτουργία του προγράμματος.

```
Accuracy: 0.957333
Precision: 0.925620
Recall: 0.941176
F1 score: 0.933333
```