FAQ Εργασιών

Για όλες τις ασκήσεις

Ορατό output στο ipynb

Η παράδοση του ipynb (το download) να γίνεται με όλα τα κελιά να έχουν τρέξει (να είναι ορατό το output του κώδικα)

Εκτέλεση κελιών κώδικα εντός notebook που απαιτούν χρόνο, χωρίς επίβλεψη (non-interactive), έχοντας κλειστό το browser

Οι οδηγίες αυτές είναι ιδιαίτερα χρήσιμες όταν θέλουμε να τρέξουμε ένα κελί κώδικα που μπορεί να πάρει πολύ χρόνο να ολοκληρωθεί.
Αρχικά, όταν κλείνουμε το tab του browser ή τον browser ο πυρήνας εξακολουθεί να εκτελείται μέχρι να λάβει εντολή τερματισμού, είτε από εμάς, είτε από το περιβάλλον του. (ref)
Αυτό που θα χάσουμε αν κλείσουμε το κελί θα είναι του output του κελιού που πιθανότατα περιέχει τα αποτελέσματα που μας ενδιαφέρουν. Το output κατευθύνεται στο stdout (standard output) του λειτουργικού και το jupyter το φέρνει μέσα στο browser.

Αυτό που μπορούμε να κάνουμε για να σώσουμε την έξοδο στο stdout είναι να βάλουμε στην αρχή του κελιού τη μαγική εντολή "capture" ως εξής:

%%capture output

η οποία θα σώσει στη μεταβλητή "output" την έξοδο του κελιού όταν αυτό ολοκληρώσει. Τρέχουμε το κελί και μπορούμε άνετα να κλείσουμε το tab ή το browser (άσχετα αν μας προειδοποιεί ο browser να μην φύγουμε).

Αν αργότερα ανοίξουμε το browser και το κελί έχει τερματίσει, με ένα απλό:

print(output)

θα πάρουμε την έξοδο του κελιού. Εάν η έξοδος περιλαμβάνει και γραφικά θα πρέπει να δώσετε:

output.show()

Προσοχή: τα περιβάλλοντα cloud αφήνουν τους πυρήνες των notebooks να τρέχουν μόνο μέχρι ένα maximum χρόνου οπότε δεν ενδείκνυται για πολύ μεγάλα πειράματα. Use at your own risk.

Αναφορές - Βιβλιογραφία - Κώδικας

Αν βρείτε πληροφορίες, αναφορές, papers στο internet σχετικά με το πρόβλημα που μελετάτε η σωστή πρακτική είναι να βάζετε την πηγή (url, βιβλιογραφική αναφορά κοκ). Το ίδιο ισχύει και με τον κώδικα. Όλοι δανειζόμαστε (αρκεί να καταλαβαίνουμε τί κάνει ο κώδικας) αλλά η βέλτιστη πρακτική είναι να βάζουμε την πηγή (σε σχόλιο), και για εμάς και για τρίτους που διαβάζουν τον κώδικα.

Άσκηση 1 Ταξινόμηση

Q: Βέλτιστες τιμές υπερπαραμέτρων στο gridsearchCV και τελική επίδοση στο test set

A: Το gridsearchCV πραγματοποιείται εντός του training set και μας δίνει τον καλύτερο συνδυασμό υπερπαραμέτρων με βάση το μέσο όρο του metric που χρησιμοποιούμε σε όλα τα folds του crossvalidation. Αυτό δεν σημαίνει πάντοτε ότι αυτές οι υπερπαράμετροι θα δώσουν τη μέγιστη δυνατή τιμή στο test set (δεν αποκλείεται πχ ένας άλλος συνδυασμός να δώσει λίγο καλύτερες τιμες - υπάρχει ένα τέτοιο παράδειγμα στα notebooks) για το metric ωστόσο πάντα θα δώσουν μια πολύ καλή τιμή. Συνεπώς, επειδή η χρήση του test set **απαγορεύεται** πλήρως στην εκπαίδευση, ο μόνος τεκμηριωμένα ορθός τρόπος για να βρίσκουμε βέλτιστες υπερπαραμέτρους είναι το gridsearchcv.

Q: Κάποια datasets είναι ήδη χωρισμένα σε train και test

A: Αν τα ποσοστά του test και του train είναι κοντά σε αυτά που ζητάει η εκφώνηση, μπορείτε να χρησιμοποιήσετε το train ως έχει για διασταυρούμενη επικύρωση πλέγματος. Αν τα ποσοστά είναι πολύ διαφορετικά, μπορείτε να τα ενοποιήσετε και διαχωρίσετε με την train_test_split. Όπως και να έχει εξηγήστε την επιλογή σας.

Q: Multiclass Classification

Σε κάποια προβλήματα οι κατηγορίες εξόδου δεν είναι μόνο δύο (binary) αλλά περισσότερες (multiclass). Υπάρχουν ταξινομητές που μπορούν να κάνουν multiclass ταξινόμηση απευθείας (inherently multiclass) και αυτοί που δεν μπορούν, στους οποίους πρέπει να μετασχηματισουμε το πρόβλημα από multiclass συνήθως σε one-vs-all (binary). Όλοι οι ταξινομητές που χρησιμοποιούμε (GNB, kNN, MLP) είναι inherently multiclass (το είδαμε στο Iris που έχει 3 κατηγορίες εξόδου) και δεν χρειάζεται να κάντε one-vs-all. Περισσότερα εδώ.

Q: Dataset specific questions

S01

Το χαρακτηριστικό εξοδου (η κλάση) είναι το 13ο “alive-at-1”. Θα κρατήσετε μόνο όσα δείγματα δεν έχουν “?” στο “alive-at-1”. Τα χαρακτηριστικά είναι τα **3 έως 9** (τα υπόλοιπα μπορούν να αγνοηθούν). Η πρόβλεψη μπορεί να γίνει και με τα χαρακτηριστικά 1-9 (όπως διατυπώθηκε αρχικά) απλά θα δίνει πολύ υψηλές τιμές (που δεν προσφέρονται για πολύ περεταίρω βελτιστοποίηση) γιατί υπάρχει μεγάλη συσχέτιση (αν και όχι απόλυτη) μεταξύ των χαρακτηριστικών 1 και 2 και της μεταβλητής εξόδου. Δείτε εδώ τη συσχέτιση Pearson μεταξύ χαρακτηριστικών και εξόδου. Οι κολόνες 1-9 είναι τα χαρακτηριστικά εισόδου (έχουμε αφαιρέσει τα 3 άχρηστα αρχικά χαρακτηριστικά) και η 10η κολόνα είναι η έξοδος. Τιμές κοντά στο -1 ή στο 1 δείχνουν υψηλή συσχέτιση, αντίστροφη (-1) ή ανάλογη (1). Δείτε εδώ μια βιβλιογραφική αναφορά για τη διαχείριση του dataset.

S11

Μπορείτε να το αναλύσετε ως binary classification παίρνοντας ως κατηγορία εξόδου μόνο το consensus.

B03

Προχωρήστε κανονικά σε CrossValidation.

B04

Παρότι η περιγραφή αναφέρει ότι υπάρχουν απουσιάζουσες τιμές, δεν υπάρχουν. Ο ακόλουθος κώδικας που ψάχνει για “NaN” δίνει “False”

```
import pandas as pd
import numpy as np

sb =
pd.read_csv("http://archive.ics.uci.edu/ml/machine-learning-databases/spambase/spambase.data",
header=None)
sb.isnull().values.any()
```

B07

Στη βιβλιογραφία θα το δείτε πιο συχνά ως 5 ξεχωριστά προβλήματα binary ταξινόμησης (χρεοκόπησε ή όχι). Στα πλαίσια της άσκησης θα ενώσουμε και τα 5 αρχεία και θα απλοποιήσουμε το πρόβλημα σε χρεοκόπησε ή όχι, ανεξάρτητα του έτους που συνέβη αυτό.

Q: F1 micro και F1 macro

Q: Οταν χρησιμοποιούμε διαφορετικά metrics μπορεί να καταλήξουμε σε διαφορετικά μοντέλα. Και τα δύο είναι δόκιμα και θέλουμε τις παραμέτρους τους. Υπάρχει η δυνατότητα το gridsearchcv να λαμβάνει υπόψη του περισσότερες μετρικές, αλλά εμάς μας ενδιαφέρουν για κάθε ταξινομητή δύο ενδεχομένως διαφορετικά μοντέλα που προκύπτουν από gridsearch πρώτα με F1 micro και μετά F1 macro.

Q: Βελτιστοποίηση αρχιτεκτονικής

A: Πέραν της εύρεσης των βέλτιστων τιμών υπερπαραμέτρων, πρέπει να βρούμε και τη βέλτιστη αρχιτεκτονική των μετασχηματιστών. Δεν είναι καθόλου υποχρεωτικό να κάνουμε όλα τα βήματα scaling, variance threshold, PCA, μπορεί να μην χρειάζεται τίποτα από αυτά για τη βέλτιστη επίδοση. Μπορεί να χρειάζεται ένα από αυτά, δύο ή και τα τρία. Συνήθως ξεκινάμε χωρίς μετασχηματιστές, να δούμε την επίδοση μόνο του ταξινομητή για να έχουμε ένα μέτρο και αρχίζουμε να προσθέτουμε μετασχηματιστές παρακολουθώντας την επίδραση τους στις επιδόσεις, πάντα με αναζήτηση πλέγματος με διασταυρούμενη επικύρωση. Το topic αυτό είναι almost duplicate με αυτό.

Q: Πώς επεξεργάζονται τα αρχεία δεδομένων τύπου .arff?

A: Ορισμός του format .arff. Για να το μετατρέψετε ένα arff σε ένα συνηθισμένο csv αρκεί να κρατήσετε μόνο τις γραμμές που δεν ξεκινάνε με “%”, “@” και δεν είναι κενές. Μπορείτε να το κάνετε manually σε ένα editor ή με ένα oneliner στο shell:
linux:
cat data.arff | grep -ve "^@\|^%" | grep -v "^[[:space:]]*$" > data.csv
windows:
findstr /BV "@" data.arff | findstr /BV "%" | findstr /V /R /C:"^[ ]*$" > data.csv

Q: Πώς κάνουμε concatenate (ενώνουμε) αρχεία δεδομένων?

Για να ενώσουμε όλα τα csv σε ένα:
linux:
cat *.csv > all.csv
windows

copy *.csv all.csv

Q: Σε κάποια datasets το delimiter δεν αποτελείται πάντα από τον ίδιο αριθμό κενών (“ “)

Σε αυτή την περίπτωση χρησιμοποιήστε στην pd.read_csv option delim_whitespace=True χωρίς να θέσετε καθόλου option delimiter. read the docs. Γενικά μπορούμε να ορίσουμε τα delimiters με full σύνταξη regular expressions που είναι πολύ ισχυρή μέθοδος (non Python specific).

Q: Σε ένα σύνολο δεδομένων, πώς διαχειριζόμαστε τυχόν τιμές χαρακτηριστικών που απουσιάζουν (missing values)?

Έστω ότι μια τιμή που απουσιάζει συμβολίζεται με “?” (χωρίς τα εισαγωγικά).

Βρίσκουμε πρώτα τον αριθμό δειγμάτων (γραμμών) που έχουν έστω μια τιμή χαρακτηριστικού που απουσιάζει:
linux
cat data.csv | grep "?" | wc -l
windows
findstr "?" data.csv | find /c /v ""

Αν το ποσοστό των δειγμάτων με τιμές που απουσιάζουν είναι σχετικά μικρό (πχ <5% του συνόλου του dataset μπορούμε να αφαιρέσουμε τα συγκεκριμένα δείγματα
linux
cat data.csv | grep -v "?" > nomissing.data.csv
windows
findstr /V "?" data.csv > nomissing.data.csv

Αν αφαιρέσουμε δείγματα με απουσιάζουσες τιμές, το κάνουμε πριν το το χωρισμό σε train και test set, πριν ή μετά την εισαγωγή του csv στο notebook.

Αν το ποσοστό είναι μεγαλύτερο, δεν θέλουμε να “θυσιάσουμε” σημαντικό μέρος των δεδομένων. Χρησιμοποιούμε το μετασχηματιστή “Imputer” του scikit learn που αντικαθιστά κάθε απουσιάζουσα τιμή χαρακτηριστικού με τη μέση τιμή (συνεχείς μεταβλητές) ή την πιο συχνή τιμή (κατηγορικές μεταβλητές) του χαρακτηριστικού στο train set.

Για να διαβάσετε ένα αρχείο csv με απουσιάζουσες τιμές που συμβολίζονται με “?” (χωρίς τα εισαγωγικά) μπορείτε να περάσετε στην read_csv των pandas την παράμετρο na_values=["?"]. Στη συνέχεια αν θέλετε για παράδειγμα να αντικαταστήσετε με τη μέση τιμή μπορείτε απλά να θέσετε imp = Imputer(strategy='mean', axis=0) , δηλαδή χωρίς την παράμετρο missing_values.

Ο μετασχηματισμός με Imputer γίνεται στην απόλυτη αρχή της προεπεξεργασίας αλλά μετά το διαχωρισμό σε train και test set. Μπορεί επίσης να χρησιμοποιηθεί σε pipeline, ως το απολύτως πρώτο βήμα, καθώς όλοι οι υπόλοιποι estimators του scikit δεν διαχειρίζονται απουσιάζουσες τιμές.

Αν υπάρχουν και μη διατεταγμένες μεταβλητές τις μετατρέπουμε σε δυαδικές αφού διαχειριστούμε τις απουσιάζουσες τιμές.

Q: random_state

Σε κάποια παραδείγματα χρησιμοποιούμε τη random_state για να έχουμε συγκεκριμένες αρχικοποιήσεις της γεννήτριας τυχαίων αριθμών. Εσείς μην την χρησιμοποιείτε στις αρχικοποιήσεις σας.

Q: Εξισορρόπηση με undersampling

Σε μεγάλα datasets είναι πιθανό η εξισορρόπηση με oversampling να μεγαλώσει πολύ το μέγεθός τους. Μπορεί να δοκιμαστεί και η μέθοδος undersampling του imblearn που μειώνει το μέγεθος του dataset κατά την εξισορρόπηση.

Q: Πως μπορούν να επιταχυνθούν τα for loops;

Θα μπορούσε να γίνει παράλληλη επεξεργασία με τη βιβλιοθήκη joblib. Εγκατάσταση εντός notebook με “!pip install --upgrade joblib” (doc).

Q: Πως βρίσκουμε ποια είναι η βέλτιστη αρχιτεκτονική Pipeline για ένα συγκεκριμένο ταξινομητή και dataset;

Η βέλτιστη αρχιτεκτονική βρίσκεται μόνο εμπειρικά. Μπορείτε να προχωρήσετε bottom up (μόνο με estimator και να προσθέτετε μετασχηματιστές) ή top down δηλαδή με όλα τα στάδια (επιλογή χαρακτηριστικών, κανονικοποίηση, εξαγωγή χαρακτηριστικών, ταξινομητής) και να κάνετε δοκιμές αφαιρώντας κάποιο μετασχηματιστή. Η σειρά των μετασχηματιστών είναι πάντα αυτή, εκτός της περίπτωσης εφαρμογής variance threshold και min max scaler όπου πρέπει πρώτα να γίνει κανονικοποίηση και μετά επιλογή χαρακτηριστικών.

Q: Συμβουλές για το grid search

- Εάν η βέλτιστη τιμή μιας αριθμητικής υπερπαραμέτρου βρίσκεται στα άκρα του διαστήματος αναζήτησης (είναι η ελάχιστη ή η μέγιστη) μετατοπίστε το διάστημα αναζήτησης γιαυτήν έτσι ώστε να τη φέρετε στη μέση του.

- Εάν ένα dataset είναι πολύ μεγάλο, θέλετε να εξερευνήσετε ένα μεγάλο χώρο αναζήτησης υπερπαραμέτρων και δείτε ότι καθυστερεί το gridsearch cv, μπορείτε να ξεκινήσετε την

αναζήτηση κάνοντας τυχαίο sampling δειγμάτων ενός ποσοστού του dataset. Εφόσον προσδιορίσετε τις περιοχές τιμών που δίνουν καλά αποτελέσματα, μπορείτε να κάνετε ένα πιο στενό grid search με όλα τα δείγματα.

- Μπορείτε να δοκιμάσετε τιμές κοντά στις προτεινόμενες αλλά και μια τάξη μεγέθους μικρότερες και μεγαλύτερες. Σε κάποια θέματα το ίδιο το scikit learn έχει προτεινόμενες προσεγγίσεις Μπορείτε επίσης να ψάξετε και στο internet ή papers. Αν χρησιμοποιήσετε εξωτερικές γνώσεις σημειώστε στο markdown το link της αναφοράς.

- Μπορείτε να χρησιμοποιήσετε τη συνάρτηση “time” (υπάρχουν παραδείγματα στα notebooks) για να μετρήσετε το χρόνο ολοκλήρωσης ενός grid search και να έχετε μια εικόνα για τους χρόνους σε μεγαλύτερους χώρους αναζήτησης.

- Δεν χρειάζεται να σώσετε όλα τα block κώδικα με τα διαφορετικά grid search, απλά κάντε αντικατάσταση των τιμών στο ίδιο block κώδικα και σημειώστε απαραίτητα τα διαδοχικά search spaces και βέλτιστες τιμές κάθε run σε markdown.

Άσκηση 2 Text mining - Ομαδοποίηση

Q: Αποθήκευση αντικειμένου SOM

Εφόσον έχετε αρχικοποιήσει και εκπαιδεύσει ένα αντικείμενο “som” με τις somoclu.Somoclu(...) και som..train(...) και αφότου υπολογίσετε και τα clusters με som.cluster(...) αποθηκεύετε ολόκληρο το αντικείμενο με joblib.dump(som, 'som.pkl').

Q: Μεγέθη αρχείων & χρόνοι εκτέλεσης

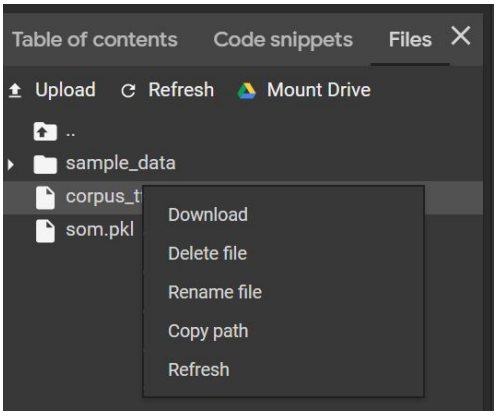
A: Ένα **συμπιεσμένο** corpus_tf_idf.pkl ενός αποτελεσματικού recommender μπορεί να είναι και κάτω από 4MB. Ένας συμπιεσμένος χάρτης 25x25 μπορεί να είναι κάτω από 3MB. Εάν το training του SOM παίρνει πολύ ώρα και τα αρχεία σας βγαίνουν υπερβολικά μεγάλα **δεν έχετε βελτιστοποιήσει τη διανυσματική αναπαράσταση των κειμένων όσο πρέπει**. Όπως αναφέρεται στην εκφώνηση *“Χρησιμοποιήστε την time για να έχετε μια εικόνα των χρόνων εκπαίδευσης. Ενδεικτικά, σε εκτέλεση στα clouds, με σωστή κωδικοποίηση tf-idf, μικροί χάρτες για λίγα δεδομένα (1000-2000) παίρνουν γύρω στο ένα λεπτό ενώ μεγαλύτεροι χάρτες με όλα τα δεδομένα μπορούν να πάρουν 10 λεπτά ή και περισσότερο.”.*
Προσπαθήστε να βελτιστοποιήσετε τους μικρούς χάρτες πρωτού δοκιμάσετε μεγαλύτερους για να μην περιμένετε πολύ.

ΠΡΟΣΟΧΗ: στο mycourses το max upload filesize του zip με τα 2 pkl (corpus και SOM) και τα αρχεία (.ipynb, py) σας είναι **29MB**.

Q: Πώς αποθηκεύονται και πως κατεβαίνουν τοπικά τα joblib pkl dumps?

A:

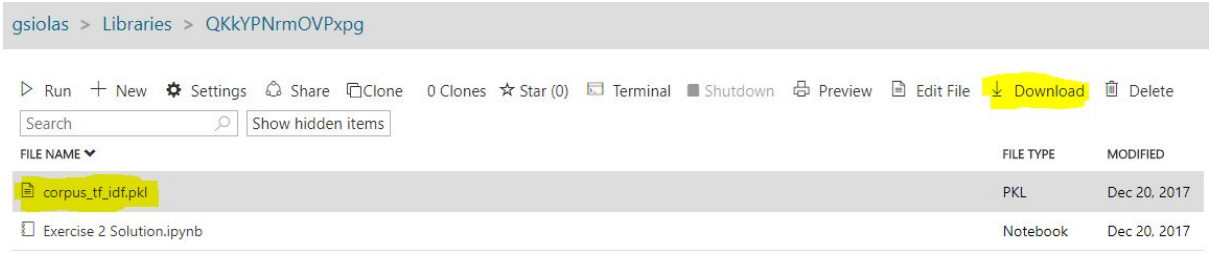
Google Colaboratory



Κάντε expand το αριστερό sidebar του notebook. Στο tab “Files” θα δείτε όλα τα αρχεία εντός του χώρου του workspace. Κάντε δεξί κλικ και “Download” σε αυτό που θέλετε να κατεβάσετε.

Microsoft Azure

Στα Microsoft Azure Notebooks εμφανίζονται ως απλά αρχεία στο library μέσα στο οποίο βρίσκεται το notebook. Μπορείτε να τα διαλέξατε και να πατήσετε download



Kaggle

Θεωρούμε ότι έχουμε δημιουργήσει τα αρχεία .pkl. Σε επόμενο κελί κώδικα γράψτε

```
from IPython.display import FileLink, FileLinks
FileLinks('.') #lists all downloadable files on server
```

και κάντε run. Θα εμφανιστεί κάτι τέτοιο:

με απλό κλικ στα ονόματα κατεβαίνουν τα αρχεία (πηγή)

Q: Εντός των κειμένων υπάρχουν κάποιοι χαρακτήρες της μορφής “\xd3”

A: Πρόκειται για την hex κωδικοποίηση unicode χαρακτήρων. Αγνοήστε το στην επεξεργασία σας καθώς είναι λίγοι σε αριθμό σε σχέση με τη συλλογή. Αν θέλετε να δείτε ποιος χαρακτήρας είναι μπορείτε να τρέξετε
print u'\xd3'
Ό
Μπορείτε επίσης να συμβουλευτείτε και ένα πίνακα unicode lookup.

Q: Recommender: ομοιότητα ή απόσταση συνημιτόνου;

A: Στην εκφώνηση της άσκησης ζητάμε να υπολογίσετε την ομοιότητα συνημιτόνου της ταινίας στόχου με όλες τις ταινίες της συλλογής, να φτιάξετε μια λίστα με φθίνουσα σειρά ομοιότητας και να επιστρέψετε τα πρώτα max_recommendations.
Στο “Lab 7 Text mining” έχουμε μιλήσει και για ομοιότητα και για απόσταση συνημιτόνου και το παράδειγμα που δίνουμε είναι με την απόσταση (sp.spatial.distance.cosine). Γενικά ισχύει το πολύ απλό “cos_similarity = 1 - cos_distance”, δηλαδή, για παράδειγμα, μια ταινία έχει ομοιότητα με τον εαυτό της 1 και απόσταση 0. Συνεπώς αν θέλετε να χρησιμοποιήσετε απόσταση συνημιτόνου στο recommender, πρέπει να φτιάξετε μια λίστα με αύξουσα απόσταση συνημιτόνου και να επιστρέψετε τα πρώτα max_recommendations.
Αν ακολουθήσετε τη μέθοδο της ομοιότητας όπως λέει η άσκηση, θα πρέπει να χρησιμοποιήσετε την cosine_similarity. Είναι προφανώς τελείως ισοδύναμες οι προσεγγίσεις.

Q: Recommender: πώς βρίσκουμε πιο εύκολα “ποια είναι η θεματική που ενώνει τις ταινίες” που επιστρέφει το recommender;

A: Χρησιμοποιήστε εντός browser το “Find” (Ctrl + F) για να κάνετε highlight τις λέξεις που εμφανίζονται στο κείμενο στόχο και στις προτάσεις του συστήματος συστάσεων. Πχ για την ίδια ταινία στόχο (Q-planes): “ship”, “aircraft”, “spy”, “factory” (εδώ είναι για τα δύο πρώτα recommendations λόγω screenshot, θα είναι παρόμοια για τα πιο κάτω recommendations). Πρόκειται λοιπόν κυρίως για ταινίες δράσης, κατασκοπικές και πολέμου στα οποία διαδραματίζουν σημαντικό ρόλο πλωτά και εναέρια μέσα.

Q: Recommender: τί εννοούμε στη βελτιστοποίηση του tf-idf με το “πολλά φαινόμενα που το σύστημα εκλαμβάνει ως ομοιότητα περιεχομένου, επί της ουσίας δεν είναι επιθυμητό να συνυπολογίζονται”.

A: Επειδή η database μας είναι περιγραφές υποθέσεων ταινιών πολύ συχνά η σύνοψη ξεκινά με φράσεις όπως “The plot of this film is about…”. Είναι φανερό ότι τα ουσιαστικά “plot” και “film” δεν αποτελούν μέρος της σημασιολογικής περιγραφής του ίδιου του φιλμ. Υπάρχουν και κάποιες ακόμα τέτοιες λέξεις. Παρόμοιο φαινόμενο παρατηρείται σε κάποιες περιπτώσεις και με κύρια ονόματα κλπ. Υπάρχει τρόπος να το διαχειριστείτε διαβάζοντας το documentation.

Άσκηση 3 Βαθιά Μάθηση

Q: Θέλουμε να χρησιμοποιήσουμε τα EfficientNets ωστόσο πετάει error “module ‘tensorflow’ has no attribute ‘random_normal’”

Ο κώδικας είναι για tf1 και μπορείτε να κάνετε αυτές τις αλλαγές για αρχή αλλά πιθανότατα θα σκάσουν και άλλες μετά. Κάντε το αν θέλετε και αν κάνετε port τα EfficientNets στο tf2, να το βάλουμε μετά στο Github, βοηθάτε την ερευνητική κοινότητα. Αλλιώς, δεν έχετε καμία υποχρέωση να χρησιμοποιήσετε τα EfficientNets.

Q: Μπορούμε να χρησιμοποιήσουμε modules απο το v1 του TensorFlow API;

A: Short answer: όχι. Υπάρχει πράγματι η λύση να δουλέψει κανείς σε εγκατάσταση tf2 σε full compatibility με κώδικα για το tf1 χωρίς καμία αλλαγή:

```
import tensorflow.compat.v1 as tf
tf.disable_v2_behavior()
```

Προφανώς αυτή τη στιγμή υπάρχουν πολλά περισσότερα διαθέσιμα έτοιμα κομμάτια κώδικα για tf1 και κάποιο μπορεί να ταιριάζει σε αυτό που θέλετε να πετύχετε. Ωστόσο, αν δεν μπορείτε ή δεν θέλετε να το υλοποιήσετε from scratch σε tf2 ενώ το έχετε βρει σε tf1, η βέλτιστη πρακτική είναι η μετατροπή του κώδικα tf1 σε tf2: Migrate your TensorFlow 1 code to TensorFlow 2

Q: Να χρησιμοποιήσουμε μόνο αρχιτεκτονικές from scratch ή μόνο transfer learning;

A: Στην εκφώνηση σας ζητείται να συγκρίνετε τις δύο προσεγγίσεις. Συνεπώς και τα δύο.

Q: Πόσες περίπου αρχιτεκτονικές δικτύων αναμένετε να βελτιστοποιήσουμε; Πόσες βελτιστοποιήσεις είναι απαραίτητες και πόσες προαιρετικές;

A: Η τρίτη και τελευταία άσκηση του μαθήματος των Νευρωνικών έχει περισσότερο ερευνητικό χαρακτήρα από τις δύο πρώτες. Επί της ουσίας, η μελέτη του συγκεκριμένου dataset με Βαθιά Μάθηση είναι ένα πρόβλημα εν πολλοίς ερευνητικό και όχι περιορισμένα εκπαιδευτικό. Συνεπώς δεν υπάρχει κάποια ποσοτική απάντηση ως προς αυτό, παρά μόνο αυτό που υπάρχει ήδη στην εκφώνηση: ξεκινήστε από διάφορες βασικές αρχιτεκτονικές και δοκιμάστε βελτιστοποιήσεις. Η βελτίωση ενός μοντέλου είναι πάντα σχετική μόνο ως προς τον εαυτό του, initial model vs optimized. Με βάση αυτή τη λογική, θα μελετήσουμε την προσπάθεια κάθε ομάδας ξεχωριστά, λαμβάνοντας υπόψη μας βέβαια τους χρονικούς περιορισμούς και τα πλαίσια του μαθήματος.

Q: Πόσα notebooks να παραδώσουμε; Να παραδώσουμε report ή notebooks;

A: ένα μόνο notebook θα ήταν το επιθυμητό. Επειδή καταλαβαίνουμε ότι θα υπάρξουν πολλές δοκιμές, για αυτό προτείνουμε να παραδώσετε ένα notebook και να συγκεντρώσετε τα αποτελέσματα και συγκριτικά σε ένα report. Αν φτιάξετε και άλλα notebooks παραδώστε τα μέσα στο zip.