Project :

Music Recommendation System using Word Emdbeddings

Ομάδα Τ : Στεργιούλης Βασίλης 03166, Παντόπουλος Εμμανουήλ 03222

26 Ιανουαρίου 2024

Το παρών έγγραφο αποτελεί την τελική αναφορά στο τελικό *Project* του μαθήματος

0.1 Abstract

Η εξέλιξη των Recommender Systems έχει προσφέρει σημαντική βελτίωση στην αντιμετώπιση του προβλήματος της πληθώρας πληροφοριών στις σύγχρονες εφαρμογές. Τα *Recommender Systems* (ελληνιστί Συστήματα Συστάσεων) , είναι αλγόριθμοι , που χρησιμευούν ως ένα ” φίλτρο ” πληροφοριών που προτείνουν αντικείμενα σε χρήστες, κάνοντας χρήση της Μηχανικής Μάθησης και της Επιστήμης Δεδομένων έτσι ώστε η πρόταση που θα γίνει εν τέλει να είναι αρεστή από τον ίδιο το χρήστη και ως αποτέλεσμα αυτό να φέρει μεγαλύτερη επιτυχία στην εφαρμογή. Γενικά οι εφαρμογές των Συστημάτων Συστάσεων είναι ευρείες και ποικίλες, καλύπτοντας διάφορα κεφάλαια της καθημερινής ζωής. Ανάμεσα σε αυτές, εντοπίζουμε τις δημοφιλείς πλατφόρμες όπως Spotify, YouTube, Amazon και άλλα, με απώτερο σκοπό των εφαρμογών αυτών να κάνουν καλύτερη την εμπειρία των ήδη υπάρχων χρηστών, αλλά και να τραβήξουν και καινούργιους χρήστες. Στο τέλος επικρατούν οι εφαρμογές αυτές με τα καλύτερα συστήματα συστάσεων στα αντικείμενά τους. Αντικείμενα όπως τραγούδια (με τα οποία ασχοληθήκαμε στην παρούσα εργασία - Project), ταινίες, βίντεο-παιχνίδια, ξενοδοχεία, υπηρεσίες μετακίνησης, φαγητά και γενικά όποιο αντικείμενο ή υπηρεσία έχει κάποιου είδους ανταγωνισμό στο τομέα του και πρέπει να γίνει κάποια πρόταση. Τα Συστήματα συστάσεων μπορούν να κατηγοριοποιηθούν ως εξής:

- **Content based , (με βάση το περιεχόμενο)**
- **Collaborate Filtering**
- **Hybrid Filtering** , μία μίξη , των παραπάνω .

Κάθε κατηγορία εξυπηρετεί διαφορετικές ανάγκες σε διαφορετικές εφαρμογές και δίνει στον χρήστη την ευκαιρία να ζήσει μια ξεχωριστή εμπειρία ανάλογα και με την κατηγορία του συστήματος συστάσεων της εφαρμογής. Η σημασία των Συστημάτων Συστάσεων σε αυτές τις εφαρμογές έγκειται στην δημιουργία μιας εξατομικευμένης εμπειρίας για τον κάθε χρήστη, προωθώντας περιεχόμενο που ανταποκρίνεται στις προτιμήσεις του. Με την συνεχή εξέλιξη των αλγορίθμων και την αξιοποίηση νέων τεχνολογιών, αναμένεται να βελτιώνονται διαρκώς οι υπηρεσίες αυτές, ενισχύοντας την ανταγωνιστικότητα των εφαρμογών που τις υιοθετούν. Επιπλέον, η εξέλιξη των Συστημάτων Συστάσεων συνδέεται στενά με την ανάπτυξη της τεχνητής νοημοσύνης και της επιστήμης δεδομένων. Οι αλγόριθμοι μηχανικής μάθησης, όπως οι αλγόριθμοι συστάσεων, εκπαιδεύονται με τη χρήση μεγάλων όγκων δεδομένων, επιτρέποντας τους να αντλούν πολύπλοκες συνδέσεις και προτιμήσεις που θα δυσκολευόταν να αντιληφθούν με κλασικές μεθόδους.

0.2 Introduction

Η τρέχουσα επανάσταση στη μουσική βιομηχανία αντιπροσωπεύει σημαντικές ευκαιρίες και προκλήσεις για τα **συστήματα σύστασης** γενικότερα. Τα συστήματα σύστασης είναι τώρα ουσιαστικά για τις πλατφόρμες μουσικής αναπαραγωγής μέσω του διαδικτύου (music streaming platforms), οι οποίες αυξάνονται γρήγορα στην ακροαματικότητα και γίνονται η κορυφαία πηγή εσόδων για τη μουσική βιομηχανία. Είναι ολοένα πιο συνηθισμένο για έναν ακροατή απλά να ακούει μουσική παρά να την αγοράζει και να την κατέχει σε προσωπική συλλογή. Σε αυτό το σενάριο, διακρίνεται ένα πολύ ευρύ φάσμα προκλήσεων, όπως η συνεχόμενη σύσταση και η σχετική σύσταση (*sequential recommendation and contextual recommendations*). Οι τεχνολογίες σύστασης επηρεάζουν τώρα όλους τους εμπλεκόμενους στον πλούσιο και πολύπλοκο κόσμο της μουσικής βιομηχανίας (ακροατές, εταιρείες δίσκων, δημιουργούς μουσικής και παραγωγούς, αίθουσες συναυλιών, διαφημιστές, κ.λπ.). Έτσι, γίνεται αντιληπτό ότι ένα καλό ή ενδιαφέρον μουσικό σύστημα σύστασης, μπορεί να έχει μεγάλη απήχηση. Για μας βέβαια που ασχοληθήκαμε με αυτό, είναι βέβαιο ότι θα αποκτήσαμε κάποια μικρή εμπειρία στον τομέα της Μηχανικής Μάθησης αλλά και πιο συγκεκριμένα στο τομέα των **Music Recommender Systems**.

0.2.1 General Rundown

Στο project αυτό μας δόθηκε η ευκαίρια να ασχοληθούμε με ένα Music Recommender System και να το προγραμματίσουμε. Μετά από αρκετή συζήτηση και έρευνα αποφασήσαμε να γράψουμε ένα Recommender System το οποίο να δέχεται σαν **input** ένα τραγούδι και να δίνει σαν **output** τα recommendation (τραγούδια) σύμφωνα με διάφορα κριτήρια. Αυτά τα κριτήρια όπως θα αναλυθούν και πιο κάτω είναι για παράδειγμα κατά πόσο είναι νοηματικά όμοια και πόσο μοιάζουν με άλλα τραγούδια σύμφωνα με δύο νουβέλες: του Dostoevsky το ”The Idiot” και του Joyce το ”Ulysses”

0.2.2 Text Processing

Αρχικά το πρόγραμμα μας αξιοποιεί δύο μεγάλες βάσεις δεδομένων, μία του Spotify (ήδη υπάρχουσα μουσική πλατφόρμα) και μία του Yahoo, επειδή κάθε βάση δεδομένων επικεντρώνεται σε διαφορετικό κομμάτι του προβλήματος μας. Στο κομμάτι της **αναζήτησης μέσω στίχων**, μας εξυπηρέτησε ιδιαίτερα η βάση δεδομένων του Spotify, αφού περιέχει πολλά μουσικά κομμάτια, αλλά και τους καλλιτέχνες και τους στίχους κάθε κομματιού. Γι αυτήν την προσέγγιση μία πρώτη ιδέα που δόθηκε, ήταν η επεξεργασία των ίδιων των στίχων (*στην οποία θα αναφερόμαστε στο υπόλοιπο ως προ επεξεργασία για χάρην συντομίας*) με κάποιο μοντέλο ***NLP*** ή αλλιώς *Natural Language Processing*. Κύριες ιδέες σε αυτό , αποτελούν τόσο το κομμάτι του *text normalization* όσο και του *text tokenization* για να εξαλείψουμε από το κείμενο χαρακτήρες που δυσκολεύουν την ανάλυση του (*χαρακτήρες αλλαγής κειμένου, τελεία, χαρακτήρες που υποδηλώνουν νέα γραμμή και τα λοιπά*), όπως και του *Word Normalization, Lemmatization και Stemming* για την σωστή επεξεργασία του . Ειδικά , όσο αφορά το δεύτερο και πιο δύσκολο μέρος, οι αλγόριθμοι που επιχειρήσαμε να υλοποιήσουμε είναι :

- **TFIDF** , ένας αλγόριθμος που χρησιμοποιεί την συχνότητα που εμφανίζονται οι λέξεις μέσα σε ένα κείμενο ή μέρος του , για να καθορίσει την σχετικότητα τους στο κείμενο
- **Word Embeddings** , η πιο εξελιγμένη μορφή του προηγούμενου και την οποία υλοποιήσαμε.

0.2.3 Similarity based on text

Όπως προδίδει και ο τίτλος , το επόμενο κομμάτι της εργασίας , θα αφορά του κατά πόσο τα αποτελέσματα της προ επεξεργασίας, σχετίζονται με το τραγούδι που έχουμε εισάγει στο σύστημα . Ίσως η επιλογή του μετρικού ομοιότητας , θα βασίζεται πάνω στην ομοιότητα συνημιτόνου ή *Cosine Similarity* . Εδώ θα θεωρήσουμε επίσης ότι τα αποτελέσματα θα είναι διανύσματα στον δισδιάστατο χώρο και όσο μικρότερη η γωνία που θα σχηματίζουν ανά δύο τα χαρακτηριστικά μας , τόσο μεγαλύτερη και η ομοιότητα τους . Να σημειωθεί , ότι οι μέχρι στιγμής διεργασίες θα γίνονται πάνω στην δεύτερη βάση δεδομένων .

0.2.4 Filtering using User Matrix

Τελευταίο μέρος της εργασίας , θα είναι η επικύρωση του προηγούμενου αποτελέσματος με βάση τον πίνακα χρηστών που μας παρέχει η βάση δεδομένων του yahoo . Επί της ουσίας , εφόσον βρεθούν τα επιθυμητά τραγούδια , το τελικό αποτέλεσμα θα αποτελείται και με διασταύρωση στον πίνακα των βαθμολογιών χρηστών . Για να επιτευχθεί το παραπάνω , θα μπορούσαμε να χρησιμοποιήσουμε μία πληθώρα αλγορίθμων πολλαπλασιασμού πινάκων , αλλά καταλήξαμε στην πιο απλή λύση , κρατώντας *την μέση βαθμολογία* κάθε καλλιτέχνη . Ενδεικτικά , μερικοί των αλγορίθμων :

- **Collaborative filtering - SVD**
- **Matrix Factorization as Gradient Descent**
- **Stochastic Gradient Descent**
- **Non-Negative Matrix Factorization**

0.3 Literature Review

0.3.1 Text Processing

Το τελικό αποτέλεσμα αυτού του Project είναι σίγουρα βασισμένο σε μεγάλο βαθμό στην διαδικτυακή έρευνα που έγινε κατά την ολοκλήρωσή του. Κάνοντας αυτήν την έρευνα σε επιστημονικά άρθρα, βιβλία, επιστημονικά blogs αλλά και άλλες πηγές διακρίναμε πολλές διαφορετικές προσεγγίσεις για την υλοποίηση ενός Music Recommender System, αλλά ταυτόχρονα διαπιστώσαμε ότι όλες οι προσεγγίσεις είχαν ένα κοινό πρόβλημα που εν τέλη αντιμετωπίσαμε και εμείς κατά την κατασκευή του Recommender System. Το πρόβλημα αυτό ήταν η έλλειψη της "ιδανικής" βάσης δεδομένων, καθώς η αρχική ιδέα που είχαμε στο μυαλό μας φάνηκε ακατόρθωτη αφού δεν μπορούσαμε να βρούμε μία μεγάλη βάση δεδομένων γνωστών κομματιών που να διέθετε ηχητικά αποσπάσματα από τα κομμάτια τα οποία περιείχε. Έτσι εμείς, όπως και η μεγαλύτερη κοινότητα που ασχολούνταν με αυτό το θέμα, προσαρμοστήκαμε στην πραγματικότητα που επικρατούσε με τις δωρεάν βάσεις δεδομένων που βρίσκονταν στη διάθεσή μας. Μετά από αυτήν την συνειδητοποίηση ξεκίνησε εντατική μελέτη για την πραγματοποίηση των μεθόδων και διαδικασιών που απαιτούσε αυτό το Project, όπως για παράδειγμα το Language και Text Processing όπου φάνηκε ιδιαίτερα χρήσιμο το βιβλίο του *Daniel Jurafsky*, **"Speech and Language Processing"**, το οποίο μας βοήθησε στην κατανόηση των κύριων αλγορίθμων για το NLP (Normalization, Logistic Regression, RNNs and LSTMs, κ.ά.).

0.3.2 General Classification Algorithms and Neural Networks

Μεγάλη βοήθεια για να αποκτήσουμε μία ισχυρή θεωρητική βάση στο αντικείμενο των αλγορίθμων ταξινόμησης, κομμάτι απαραίτητο στα συστήματα συστάσεων αλλα και γενικότερα στη Μηχανική Μάθηση, ήταν το βιβλίο **"Pattern Recognition"** των *K.Koutroumbas , S. Theodoridis*, καθώς έχει έντονα επεξηγηματικό χαρακτήρα και αυτό μας βοήθησε να επιλέγουμε πάντα τον κατάλληλο και πιο ταιριαστό αλγόριθμο για να υλοποιήσουμε το κάθε κομμάτι του συστήματος. Χαρακτηριστικοί αλγόριθμοι που χρησιμοποιήθηκαν ήταν οι "Linear SVM Model, Cosine Similarity, K-Nearest Neighbors κ.ά.". Επιπροσθέτως, στο κομμάτι των νευρωνικών δικτύων για μια πιο ξεκάθαρη προσέγγιση στο δύσκολο αυτό κεφάλαιο της Μηχανικής Μάθησης, μελετήθηκε εξονυχιστικά το βιβλίο του *Κωνσταντίνου Διαμαντάρα* **"ΤΕΧΝΗΤΑ ΝΕΥΡΩΝΙΚΑ ΔΙΚΤΥΑ"**, με αποτέλεσμα την εισαγωγή ενός σχετικά μικρού νευρωνικού δικτύου στο τελικό σύστημα συστάσεων. Τέλος, για την κατανόηση του **Keras** που αποτελεί μεγάλο κομμάτι της βαθιάς εκμάθησης και των νευρωνικών δικτύων, εκτός από σε πληθώρα βίντεο στο You-Tube, αρκετός χρόνος καταναλώθηκε και στη μέλετη του βιβλίου **"Learn Keras for Deep Neural Networks/ A Fast-Track Approach to Modern Deep Learning with Python"** του Jojo Moolayil που μας βοήθησε στη εμβάθυνση του **Keras** καθώς και της βαθιάς εκμάθησης.

0.3.3 Collaborative Filtering

Για την γενικότερη κατασκευή ενός content-based collaborative filtering recommender system υπάρχει άφθονο υλικό στον ιστό, από πτυχιακές εργασίες μέχρι και επαγγελματικά συστήματα συστάσεων, έχοντας έτσι ποικίλους τρόπους να προσεγγίσουμε το θέμα, επιλέξαμε να ακολουθήσουμε αυτούς που φαίνονταν πιο κατανοητοί και είχαν μεγαλύτερο βάθος στην εξήγηση των συγκεκριμένων μεθοδολογιών. Έτσι, μέσα από το άρθρο **"Item-Based Collaborative Filtering in Python"** που δημοσιεύθηκε στο **"Towards Data Science"** και μιλούσε για τη κατασκευή ενός πρόφιλ ανά χρήστη αλλά και το πώς μέσα από το cosine similarity και το KNN μπορούμε να προβλέψουμε βαθμολογίες αντικειμένων χρησιμοποιώντας μία ξεκάθαρη φόρμουλα, καταφέραμε να κατανοήσουμε αυτό το βασικό κομμάτι των συστημάτων συστάσεων. Παρόμοιο περιεχόμενο είχε επίσης ένα δημοσιευμένο notebook από τον υπεύθυνο καθηγητή με τίτλο **"Recommendation Systems using Different Methods.ipynb - Colaboratory"** το οποίο επεκτείνονταν περαιτέρω στις διαφορετικές τεχνικές για collaborating, content based, και hybrid filtering. Τέλος με την απόκτηση γνώσεων πάνω στο αντικείμενο του collaborative filtering, ήταν πλέον εύκολο να τις εντάξουμε στο σύστημα συστάσεων που κατασκευάσαμε.

0.3.4 Actual Programming

Από τα προαναφερθέντα, το ισχυρό θεωρητικό υπόβαθρο, που ένα τέτοιο project απαιτεί, ήταν πλέον γεγονός, οπότε, με βοήθεια την φαντασία μας θα περίμενε κανείς πως η κατασκευή ενός αποτελεσματικού μουσικού συστήματος συστάσεων θα ήταν εύκολη υπόθεση. Ωστόσο, οι πολλές ώρες "μάχης" με τον compiler (χαριτολογώντας αναφερόμενοι στα bugs , καθώς η python δεν διαθέτει έναν), η αμέτρητες αποτυχημένες δοκιμές, οι ατελείωτες κακές συστάσεις που παράχθηκαν και οι μικρές διαφοροποιήσεις που πραγματοποιούσαμε μετά από κάθε επανάληψη του προγράμματος, έρχονται να αντικρούσουν αυτή την αρκετά επιφανειακή αντίληψη. Βέβαια η ευρεία προγραμματιστική κοινότητα δεν μας απογοήτευσε, καθώς όλες οι δυσκολίες που αντιμετωπίζαμε, φάνηκαν εν' τέλη επιλύσιμες και οι λύσεις τους προφανείς, μετά από μία καλή αναζήτηση στις κορυφαίες προγραμματιστικές πλατφόρμες όπως το GitHub και το Stack Overflow. Πλατφόρμες οι οποίες με σωστή χρήση γίνονται τρομερά όπλα για κάθε προγραμματιστή, εφόσον η κοινότητα που έχει δημιουργηθεί από αυτές είναι άτομα τα οποία έχουν αντιμετωπίσει παρόμοια προγραμματιστικά προβλήματα με διαφορετικούς τρόπους.

0.4 Datasets et Features

Κύριος στόχος της εργασίας , όπως και ο σκοπός του *Recommender System* , είναι η σύσταση τραγουδιών , με βάση των στίχο. Στα *Datasets* , χρησιμοποιήσαμε δύο διαφορετικές τεχνικές (προ-)επεξεργασίας , λόγω της φύσης του καθενός . Στα *Spotify* και ενός εκ των *Yahoo* , για να μπορέσουμε να "δημιουργήσουμε" μια αντιστοιχία μεταξύ τους , εφαρμόσαμε *Stemming* , μίας τεχνικής κανονικοποίησης κειμένου (άλλες μορφές: **Lemmatization , Wordform**), στα ονόματα των καλλιτεχνών κρατώντας ατόφια τα προθέματα των ονομάτων τους .Ειδικά σε ό,τι αφορά τα *Datasets* του *Yahoo* (, βλέπε **Σχήμα 4.1**), που περιείχαν κυρίως αριθμητικές τιμές , όπως οι βαθμολογίες των χρηστών (μεταξύ τις κλίμακας του 0-100) , χρησιμοποιήσαμε τον *MinMaxScaler* , ώστε να τα μεταφέρουμε στην κλίμακα του [0,1] και κάναμε mappping τα *id* των καλλιτεχνών στο διάστημα [0, 6314] , διότι (όπως θα αναφέρουμε και αργότερα στην αρχιτεκτονική του νευρωνικού μας δικτύου) αν και υπήρχαν 6314 διαφορετικές τιμές , αυτές ήταν στο διάστημα [1000125 , 1100023] και δημιουργούσαν αρκετά (υπολογιστικά) προβλήματα στο δίκτυο μας , καθώς θα έπρεπε να σπαταλήσουμε αρκετό χώρο στο μηχάνημα μας , όπως και χρόνο για την εκπαίδευση του. Ως τελευταίο μέτρο της επεξεργασίας του , το μετασχηματίσαμε σε ένα Pivot Table , για την πιο εύκολη διαχείριση του . Τέλος στο νευρωνικό δίκτυο και για την εκπαίδευση του χωρίστηκε σε 80/20 training και validation/testing split .

Στο *Dataset* του *Spotify* (ένος υποσυνόλου του **Million Song Dataset**) που περιείχαν τους στίχους , τον κύριο κορμό της εργασίας μας , χρησιμοποιήσαμε τις εξής τεχνικές :

- Καθάρισμα κειμένου , όπως παραδείγματος χάρην από ASCII χαρακτήρες , διαγραφή "stopwords" ,δηλαδή λέξεων που εμφανίζονται αρκετά συχνά μέσα στο κείμενο , όπως αντωνυμίες και άρθρα , σαν στάδιο του **Text Preproccssing**
- **Word2Vec Processing** , στο πλέον καθαρό κείμενό μας (, βλέπε **Σχήμα 4.2**), ώστε να εξάγουμε τα features μας και να μετατρέψουμε τα λυρικά του , σε ένα διακοσαδιάστατο πλέον διάνυσμα .

Τα *Word Embdeddings* (ελληνιστί ενσωματώσεις/ενσωματωμένες λέξεων/λέξεις) είναι μια τεχνική όπου μεμονωμένες λέξεις αναπαρίστανται αριθμητικά (ως ένα διάνυσμα). Η κάθε λέξη αντιστοιχίζεται σε ένα διάνυσμα, το οποίο δημιουργείται με τρόπο που μοιάζει με ένα νευρωνικό δίκτυο[3]. Τα διανύσματα προσπαθούν να συλλάβουν διάφορα χαρακτηριστικά αυτής της λέξης σε σχέση με το συνολικό κείμενο. Αυτά τα χαρακτηριστικά μπορεί να περιλαμβάνουν τη σημασιολογική σχέση της λέξης, τους ορισμούς, το πλαίσιο καθώς και άλλα. Με άλλα λόγια το *Word to Vec* (Word2Vec) μοντέλο , που δημιουργεί τα παραπάνω embeddings , προσπαθεί να μαντέψει την ακολουθία των λέξεων , δεδομένης μίας λέξης. Στο Project , χρησιμοποιήθηκε αρχιτεκτονική *skipgram* μοντέλου που υπολογίζει την a posteriori πιθανότητα εύρεσης λέξης , σε μία συγκεκριμένη θέση του κειμένου.

Τέλος , χρησιμοποιήσαμε 2 βιβλία : *τον Ηλίθιο του Ντοστογιέφσκι και τον Οδυσσέα του Τζέιμς Τζόυς* , έτσι ώστε να μπορέσουμε να δώσουμε την ψευδαίσθηση , ότι κάποιο τραγούδι ταιριάζει περισσότερο με τα λεγόμενα του ενός (Ντοστογιέφσκι) ή του άλλου (Τζόυς) και η αναζήτηση των παρόμοιων τραγουδιών να μην γίνεται πλέον μόνο μέσω στίχου , αλλά και με βάση το "νόημα" του ενός από τα δύο βιβλία . Για την επεξεργασία τους , χρησιμοποιήθηκαν τεχνικές ίδιες με αυτές του *Spotify* .

Τα *Datasets* που χρησιμοποιήθηκαν ήταν τα εξής :

- *Spotify Million Song Dataset* , για την αναζήτηση παρόμοιων τραγουδιών με βάση των στίχο ,
  πηγή :`https://www.kaggle.com/datasets/notshrirang/spotify-million-song-dataset/data`
- *Yahoo! - Movie, Music, and Images Ratings Data Sets* , που χρησιμοποιήσαμε 2 εκ των πολλών *Datasets* τα: **ydata-ymusic-artist-names-v1-0.xlsx** για την εξαγωγή των id's των καλλιτεχνών , μαζί με τα ονόματα τους και του **ydata-ymusic-user-artist-ratings-v1-0.xlsx** για την εξαγωγή των βαθμολογιών των καλλιτεχνών από 1000 διαφορετικούς χρήστες
  πηγή: `https://webscope.sandbox.yahoo.com/catalog.php?datatype=r` (*επιλογή R2*)
- Τα δύο βιβλία μας *Fyodor Dostoyefski : The Idiot , James Joyce : Ulysses* που χρησιμοποιήθηκαν για την ταξινόμηση των τραγουδιών ως τραγούδι-Ντοστογιέφσκι, τραγούδι-Τζόυς
  πηγή: `http://wedophones.com/Manuals/Novels/The%20Idiot%20-%20Fyodor%20Dostoyevsky.pdf` `https://web.itu.edu.tr/inceogl4/modernism/Ulysses.pdf`

	anonymous_user_id	artist_id	rating
0	1	1000125	90
1	1	1006373	100
2	1	1006978	90
3	1	1007035	100
4	1	1007098	100

	artist_id	artist_name
0	-100	Not Applicable
1	-99	Unknown Artist
2	1000001	Bobby "O"
3	1000002	Jimmy "Z"

Σχήμα 4.1

0	abba	Ahe's My Kind Of Girl	Look at her face, it's a wonderful face \r\nA...	look face wonderful face means something speci...
1	abba	Andante, Andante	Take it easy with me, please \r\nTouch me gen...	take easy me please touch gently like summer e...
2	abba	As Good As New	I'll never know why I had to go \r\nWhy I had...	i ll never know go put lousy rotten show boy t...
3	abba	Bang	Making somebody happy is a question of give an...	making somebody happy question give take learn...
4	abba	Bang-A-Boomerang	Making somebody happy is a question of give an...	making somebody happy question give take learn...

Σχήμα 4.2 : μη καθαρό και καθαρό κείμενο

0.5 Methods

0.5.1 Embedding Neural Network

Ένα *Embedding Neural Network* (**EmNN**) είναι μια αντιστοίχιση μιας διακριτής — κατηγορικής — μεταβλητής σε ένα διάνυσμα συνεχών αριθμών. Στο πλαίσιο των νευρωνικών δικτύων, οι ενσωματώσεις είναι χαμηλών διαστάσεων, μαθησιακές συνεχείς διανυσματικές αναπαραστάσεις διακριτών μεταβλητών. Οι ενσωματώσεις νευρωνικών δικτύων είναι χρήσιμες επειδή μπορούν να μειώσουν τη διάσταση των κατηγορικών μεταβλητών και να αναπαριστούν με νόημα κατηγορίες στον μετασχηματισμένο χώρο.Τα **EmNN** έχουν 3 κύριους σκοπούς: (a)Εύρεση πλησιέστερων γειτόνων στο χώρο ενσωμάτωσης. (b)Αυτά μπορούν να χρησιμοποιηθούν για την υποβολή προτάσεων με βάση τα ενδιαφέροντα των χρηστών ή τις κατηγορίες συμπλέγματος.(c)Ως είσοδος σε ένα μοντέλο μηχανικής εκμάθησης για μια εποπτευόμενη εργασία. Ο αλγόριθμος που χρησιμοποιήθηκε για την εκπαίδευση του **EmNN** είναι ο *AdamW*[6] . Ο *AdamW* είναι μια μέθοδος στοχαστικής βελτιστοποίησης (η εξέλιξη του αλγορίθμου **Adam**) που τροποποιεί την τυπική εφαρμογή της μείωσης βάρους του Adam λόγω των προβλημάτων σύγκλισης του αποζευγνύοντας (*decoupling*) τη μείωση του βάρους (*weight decay*) από τις ενημερώσεις κλίσης (*gradient updates*) .Όσο αφορά τον Adam , αποτελεί έναν επαναληπτικό αλγόριθμο βελτιστοποίησης που μπορεί να θεωρηθεί ως ο συνδυασμός των **RMSPROP και SGD ή Stohastic Gradient Descent**.

0.5.2 PCA

Κύριο μέλημα του μετασχηματισμού *Karhunen-Loève* ή αλλιώς **PCA** (*PrincipalComponent Analysis*) είναι η μείωση τις διάστασης των χαρακτηριστικών και ιδανικά , η μείωση σε μία διάσταση. Ο αλγόριθμός του περιγράφεται ως εξής[2] : (I) Δημιουργία του πίνακα X , που περιλαμβάνει όλα τα διανύσματα των βαθμολογιών . (II) Υπολογισμός της Μέσης τιμής του πίνακα X και η αφαίρεση της από όλα τα στοιχεία του πίνακα X , ώστε να μπορέσουμε να « κεντράρουμε » όλα τα σημεία γύρω από το (0, 0) . (III) Ο υπολογισμός του πίνακα συνδιασποράς του X (αντί για του πίνακα Rx ή αλλιώς του πίνακα αυτοσυσχέτισης , καθώς η διαφορά μεταξύ των πινάκων είναι ελάχιστη) . (IV) Υπολογισμός των ιδιοτιμών (eigenvalues) και των ιδιοδιανυσμάτων (eigenvectors) του πίνακα X και η επιλογή της μέγιστης ιδιοτιμής . (V) Επιλογή των ιδιοδιανυσμάτων της μέγιστης ιδιοτιμής και η δημιουργία του πίνακα μετασχηματισμού A . (VI) Δημιουργία του πίνακα που περιέχει τα μετασχηματισμένα δεδομένα επάνω στο επίπεδο της προβολής (Wpca). (VII) Επιλογή του κάθετου διανύσματος στο επίπεδο της προβολής και μείωση της διάστασης των στοιχείων. Τέλος , οφείλουμε να αναφέρουμε ότι η μείωση των χαρακτηριστικών με την μέθοδο του *PCA* είναι μία μη επιβλεπόμενη διαδικασία (unsupervised) που στην προβολή των χαρακτηριστικών « κυριαρχεί » η μέγιστη ιδιοτιμή και τα ιδιοδιανύσματα που σχετίζονται με αυτήν . Τα επιλεγομένα τελικά χαρακτηριστικά , είναι όλα όσα βρίσκονται πριν την τελική "εξομάλυνση της ευθείας" .Η χρήση της παραπάνω μεθόδου κρίνεται αναγκαία για την συνέχεια της εργασίας , διότι μειώνει τον αριθμό των χαρακτηριστικών από 1000 (θεωρώντας ως χαρακτηριστικό τις προβλεπόμενες βαθμολογίες των χρηστών για τον κάθε καλλιτέχνη) σε 4 (λόγω του Cumulative Proportion) .

0.5.3 Word2Vec : Negative Sampling algorithm

Το μόνο κομμάτι που δεν καλύφθηκε στα *Word Embeddings*[3] και γενικότερα στο μοντέλο *Word2Vec* , είναι αυτό του αλγορίθμου τελικής απόφασης των βαρών , του : *Negative Sampling algorithm* (εδώ **NegS**) . Με λίγα λόγια, ορίζοντας μια νέα *objective function*, ο **NegS** αλγόριθμος στοχεύει στη μεγιστοποίηση της ομοιότητας των λέξεων στο ίδιο πλαίσιο και στην ελαχιστοποίηση της όταν εμφανίζονται σε διαφορετικά νοήματα. Ωστόσο, αντί να κάνει την ελαχιστοποίηση για όλες τις λέξεις του λεξικού εκτός από τα συμφραζόμενα, επιλέγει τυχαία μια "χούφτα" λέξεων ($2 \leqslant k \leqslant 20$) ανάλογα με το μέγεθος της εκπαίδευσης και τις χρησιμοποιεί για να βελτιστοποιήσει τον στόχο. Επιλέγουμε μεγαλύτερο k για μικρότερα σύνολα δεδομένων και αντίστροφα. Εδώ το k = 4 (gensim Default) Η *Objective function*:
$J_{ns} = -log\sigma(u_{c-m+j}^T * \vartheta_c) - \sum_{k=1}^{K} log\sigma(-\hat{u_k^T} * \vartheta_c)$, όπου :
u_c είναι τα δείγματα των λέξεων , $\hat{u_k}$ είναι τα αρνητικά δείγματα , το ϑ είναι ο μέσος όρος των *Embeddings* , m, j, K είναι δείκτες , με τον τελευταίο να δείχνει τον μέγιστο αριθμό των λέξεων στο μοντέλο .

0.5.4 SVM for Book-Song Classification

Όσο αφορά την ομοιότητα μεταξύ τραγουδιών και βιβλίων , την αντιμετωπίσαμε ως ένα πρόβλημα ταξινόμησης , μεταξύ δύο κλάσεων με χαρακτηριστικά στον πολυδιάστατο χώρο: Της *κλάσης Ντοστογιέφσκι και της Τζόυς* , με τα τραγούδια να αποτελούν τα *testing* σημεία μας . Το πρόβλημα που έρχονται να επιλύσουν οι **SVM** (*Support Vector Machines*) ταξινομητές , είναι αυτό της εύρεσης της βέλτιστης επιφάνειας ταξινομητών . Η υπόθεση μας στην εργασία , ήταν ότι οι δύο κλάσεις , ήταν γραμμικά διαχωρίσιμες στον πολυδιάστατο χώρο (*στοίχημα που τελικά απέδωσε*) , καθώς ο ταξινομητής κατάφερε ποσοστό επιτυχίας εκατό

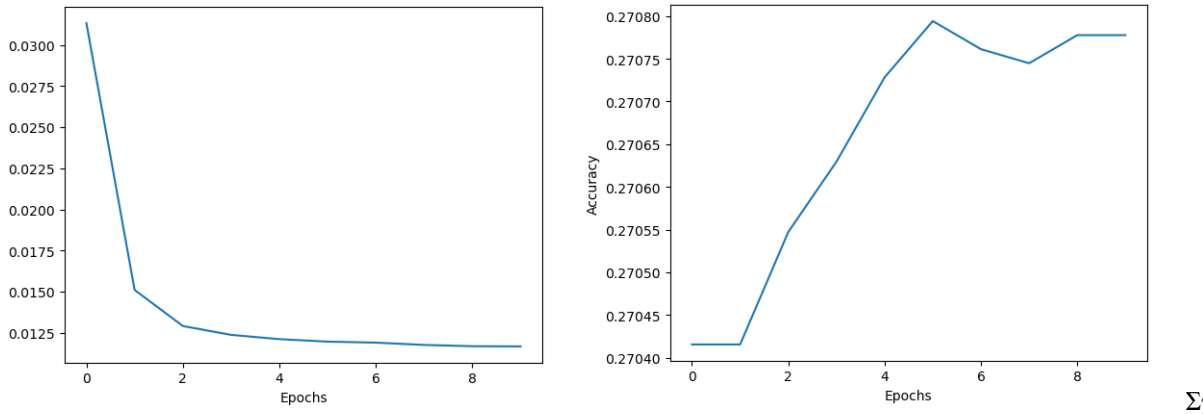
τις εκατό (το classification report , όπως και το confusion matrix και γενικότερα η όλη εκπαίδευση των **SVM** εμπεριέχονται στην επόμενη ενότητα) . Γενικότερα , τα **SVM**[1] είναι αποτελεσματικά σε χώρους υψηλών διαστάσεων και είναι ιδιαίτερα χρήσιμα όταν ο αριθμός των χαρακτηριστικών υπερβαίνει τον αριθμό των δειγμάτων,αν και είναι ευαίσθητο σε ακραίες τιμές (*outliers*) καθώς μπορούν να επηρεάσουν σημαντικά τη θέση του υπερεπίπεδου. Ακόμα , τα διανύσματα υποστήριξης (*support vectors*) είναι τα σημεία-δεδομένα που βρίσκονται πιο κοντά στο υπερεπίπεδο (διαχωρισμού) και από αυτά κρίνεται το τελικό όριο απόφασης. Τέλος η παράμετρος κανονικοποίησης στα *Support Vector Machines* (*C* ή αλλιώς *Penalty*) ελέγχει την αντιστάθμιση μεταξύ της επίτευξης ενός ομαλού ορίου απόφασης και της σωστής ταξινόμησης σημείων εκπαίδευσης .

0.5.5 Cosine Similarity

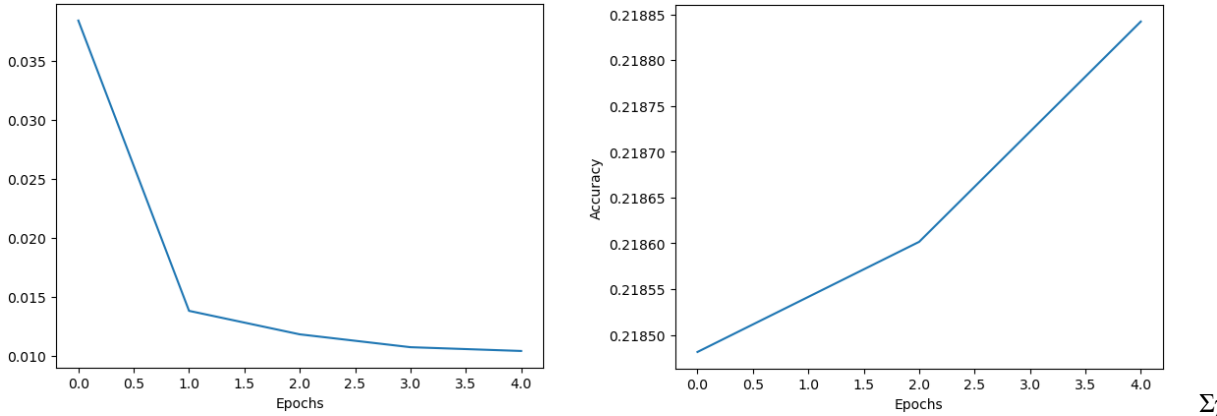
Η ομοιότητα συνημιτόνου είναι μια μέθοδος μέτρησης της διαφοράς μεταξύ δύο μη μηδενικών διανυσμάτων ενός εσωτερικού χώρου γινομένων. Μαθηματικά, μετρά το συνημίτονο της γωνίας μεταξύ δύο διανυσμάτων που προβάλλονται σε έναν πολυδιάστατο χώρο. Όσο μικρότερη είναι η γωνία, τόσο μεγαλύτερη είναι η ομοιότητα του συνημιτόνου και τόσο πιο ”όμοια” τα διανύσματα μεταξύ τους. Στην περίπτωσή μας το χρησιμοποιούμε για την εύρεση τραγουδιών (τα αντικείμενα μας , μετά την διανυσματοποίση τους μέσω του *Word2Vec*) που είναι παρόμοια με το τραγούδι που ψάχνει ο χρήστης . Η εξίσωση της απόστασης συνημιτόνου :
$\cos(\theta) = \dfrac{A \cdot B}{\|A\|_2 \|B\|_2}$, όπου $\|A\|_2$ η δεύτερη νόρμα του διανύσματος A .

0.6 Experiments/Results/Discussion

Το πρώτο πρόβλημα που καλούμαστε να διαχειριστούμε με την χρήση του **EmNN** , είναι η πρόβλεψη της βαθμολογίας κάθε καλλιτέχνη από κάθε χρήστη , πράγμα που το αντιμετωπίζουμε ως ένα πρόβλημα οπισθοδρόμησης/παλινδρόμησης (*Regression Problem*) και όχι ως ταξινόμησης . Χάρη σε αυτόν ακριβώς το διαχωρισμό , πλέον δεν μας ενδιαφέρει τόσο πολύ η ακρίβεια των αποτελεσμάτων του νευρωνικού μας δικτύου , αλλά η συνάρτηση κόστους και ειδικότερα η μείωσή της . Πέραν αυτού , το **EmNN** , καταπολεμά άθελα του και ένα ακόμα πρόβλημα , αυτό του κενού (**Sparse**) πίνακα μας , καθώς η επιλογή είτε να αφαιρέσουμε τις στήλες με τις **NaN** τιμές ή να τις εξισώσουμε με το μηδέν , δεν είναι λύσεις που θεωρούμε αποδεκτές .Για την κατάλληλη επιλογή του *optimizer* μας , (αν και εν τέλει καταλήγουμε στον δεύτερο , όπως έχει προλεχθεί) , δοκιμάστηκαν οι *Adam* και *AdamW* , με *learning rate* $= 0.007$ και ειδικότερα για τον *AdamW* με $\beta_1 = 0.9, \beta_2 = 0.99$ και $ema\text{-}momentum = 0.99$, με την τελευταία να αποτελεί μια εκθετική μορφή ομαλοποίησης , του τελικού αποτελέσματος . Μετά από σύγκριση των δύο αλγορίθμων (, βλέπε **Σχήμα 6.1** για τον *Adam* και **Σχήμα 6.2** για τον *AdamW*) και για 10 εποχές-επαναλήψεις , αν και ο *Adam* , μας δίνει το μεγαλύτερο ποσοστό ακρίβειας , ο AdamW και στις μισές ακόμη επαναλήψεις, έχει κατά πολύ μικρότερο σφάλμα , οπότε προβαίνουμε και στην επιλογή του .

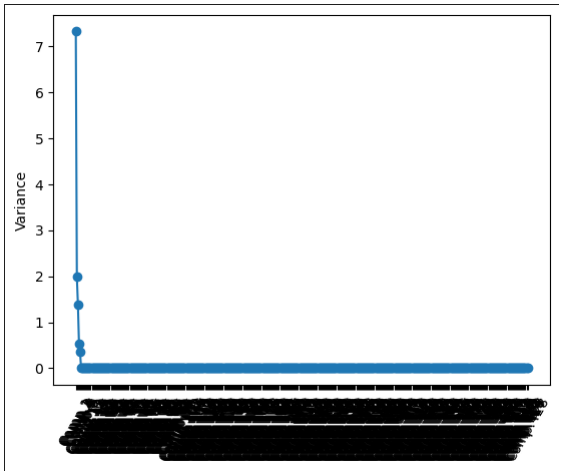


Σχήμα 6.1



Σχήμα 6.2

Αφού προβλέψουμε , την κάθε τιμή και γεμίσουμε τον τέως κενό (*Sparse*) πίνακα βαθμολογιών μας , αμέσως επόμενο βήμα είναι η μείωση των χαρακτηριστικών του πίνακα μέσω τις **PCA** ή *Principal Component Analysis* . Για την επιλογή των τελικών χαρακτηριστικών , χρησιμοποιούμε είτε το κριτήριο *Kaiser* , ή επιλέγουμε τα χαρακτηριστικά με βάση το *Cumulative Proportion* . Και οι δύο τεχνικές επιλογής χαρακτηριστικών , μας δίνουνε παρόμοια αποτελέσματα . Πριν όμως την εφαρμογή τους , καλό θα ήτανε να οπτικοποιήσουμε (*visualise*) τα αποτελέσματα της **PCA** (ή αλλιώς *Principal Components* , ***PC's***), χρησιμοποιώντας την διασπορά των *Κύριων Χαρακτηριστικών , PC's* , όπως φαίνεται και από το **Σχήμα 6.3** . Με βάση το **Σχήμα** (όσο αυτό είναι εμφανές , λόγω των πολλών χαρακτηριστικών , ευτυχώς που το κάθε **PC** απεικονίζεται ως μπίλια στο γράφημα) , προβλέπουμε ότι μάλλον τα χαρακτηριστικά , που θα κρατήσουμε , θα είναι ή τα πρώτα τέσσερα , ή τα πρώτα πέντε . Με βάση το **Σχήμα 6.4** , που εμφανίζονται τα τελικά **PC's** μετά από την χρήση του κριτηρίου *Kaiser* , όσο και του *Cumulative Proportion* , τελικά ακολουθούμε το δεύτερο τρόπο επιλογής και καταλήγουμε να επιλέξουμε συνολικά τέσσερα χαρακτηριστικά . Τέλος , ως τελικό μέτρο για την σύγκριση τν καλλιτεχνών μεταξύ τους , κρατάμε τον μέσο όρο όλων των **PC's**



Σχήμα 6.3

```
pcs_KEPT = summary.cumprop

pcs_KEPT = pcs_KEPT[pcs_KEPT['Cumulative Proportion'] < 0.99]

pcs_KEPT
```

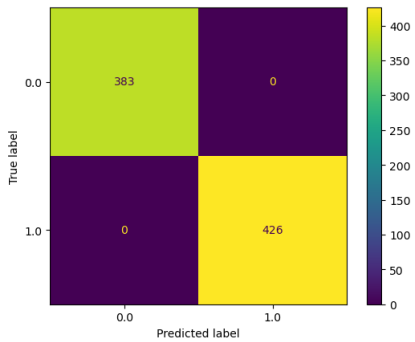
	Cumulative Proportion
PC1	0.632788
PC2	0.804207
PC3	0.923303
PC4	0.969482

```
f = summary.sdev**2

f = f[f["Standard deviation"] > 1]

f
```

	Standard deviation
PC1	7.313902
PC2	1.981301
PC3	1.376539

Σχήμα 6.4

Τελευταίο σημείο των πειραμάτων της εργασίας , πριν το τελικό αποτέλεσμα της "σύστασης" των τραγουδιών , είναι η επιλογή ενός σωστού ταξινομητή , για την τοποθέτηση των τραγουδιών στην *κλάση Ντοστογιέφσκι ή στην Τζόυς* . Η απευθείας σκέψη μας , ήταν η χρήση **SVM's** και συγκεκριμένα με *Γραμμικό Πυρήνα* (*Linear Kernel*)[5] . Μετά την επεξεργασία των βιβλίων με την χρήση του **Word2Vec** , θέτουμε σε αυτό της κλάσης Ντοστογιέφσκι την "ετικέτα" (*label*) 0 και σε αυτήν του Τζόυς την 1 . Στην συνέχεια χωρίζουμε τα δεδομένα μας σε σετ εκπαίδευσης και δοκιμής (*training et testing split*) με αναλογία 67 προς 33 και θέτουμε στο **SVM** την παράμετρο κανονικοποίησης C ίση με 100 λαμβάνοντας 100 τις 100 ποσοστό ταξινόμησης (, βλέπε **Σχήμα 6.5**). Το προηγούμενο μας δίνει την ευκολία να προχωρήσουμε στην ταξινόμηση των τραγουδιών μας(, βλέπε **Σχήμα 6.6**) που όμως ,λόγω αδυναμίας υπολογιστικής δύναμης , χρησιμοποιούμε ένα δείγμα του αρχικού *Dataset* , ίσο με 10.000 τραγούδια .



	precision	recall	f1-score	support
0.0	1.00	1.00	1.00	370
1.0	1.00	1.00	1.00	439
accuracy			1.00	809
macro avg	1.00	1.00	1.00	809
weighted avg	1.00	1.00	1.00	809

Σχήμα 6.5

```
Songs resembling Dostoyefski:  8993
Songs resembling Joyce:  1007
```

Σχήμα 6.6

Τέλος, για να βρούμε τραγούδια παρόμοια , με αυτό που θα ζητάει κάθε φορά ο χρήστης , χρησιμοποιήσαμε όπως προαναφέραμε την ομοιότητα συνημιτόνου , παράδειγμα της οποίας φαίνεται στο **Σχήμα 6.7**

```
final_recommender("The Rhythm Divine")
```

```
The current song resembles Dostoyevski
Our top 20 recommendations are:
```

	song	artist	Class
1702	Rhythm Divine	enriqu iglesia	0.0
6795	Hand On Heart	queensrych	0.0
3195	Ghost	indigo girl	0.0
3533	Looking At You	dave matthew band	0.0
6370	Lover Of My Soul	ami grant	0.0
1918	Coins In A Fountain	passeng	0.0
4071	In A Heart Like Mine	randi travi	0.0
8472	Somewhere Somehow	ami grant	0.0
3236	The Real Thing	faith no more	0.0
3537	Slow Jams	quinci jone	0.0
4824	Begin The Beguine	perri como	0.0
2525	The Way That You Love	vanessa william	0.0
5873	Carnival	tori amo	0.0
6746	The Rose	leann rime	0.0
737	Giving Up Hurts The Most	underoath	0.0
2216	Rhythm Of Hope	queensrych	0.0
8866	How Do I Love Thee	queen latifah	0.0
567	Scanner	gari numan	0.0
248	The Rose	jani joplin	0.0
473	Stompin' At The Savoy	judi garland	0.0

Σχήμα 6.7 , το τελικό αποτέλεσμα του Recommender System

0.7 Conclusion

Γενικά , υπάρχουν αρκετοί διαφορετικοί τρόποι για την δημιουργία ενός Συστήματος Συστάσεων . Μπορεί επί παραδείγματι , η διανυσματοποίηση των λέξεων να γινόταν με **TF-IDF** , αντί αυτού που χρησιμοποιήθηκε στην εργασία , γεγονός όμως , που θα μείωνε την απόδοση τόσο του **SVM** , όσο και της *Cosine Similarity*. Ένα ακόμα καλό βήμα , για να μπορέσει να γίνει ακόμα πιο ακριβές το παραπάνω σύστημα , θα ήταν όχι μόνο η ανάλυση μέσω των στίχων , αλλά και η εξαγωγή των φασματικών ή *cepstral* και προσωδικών χαρακτηριστικών της μελωδίας ενός τραγουδιού , όπως παραδείγματος χάρην η χροιά (*timbre*) , αλλά και των διάφορων αρμονικών (*Harmonics* , που χαρακτηρίζονται κυρίως από την συχνότητα F_0 του σήματος) . Ένα ακόμα μέρος που θα μπορούσε να προστεθεί στην εργασία , είναι η ταξινόμηση με πολλαπλά βιβλία . Βέβαια , το δύσκολο με το συγκεκριμένο εγχείρημα , είναι το πώς θα βρεθούνε τα βιβλία , όπως και η μορφή τους , που θα επιφέρουν αρκετές δυσκολίες στο κομμάτι της προ-επεξεργασία τους .

0.8 Contibutions

Emmanouil Pantopoulos

- Cosine Similarity
- PCA
- TF-IDF (1st Idea for implementation)
- Collaborative Filtering
- Θεωρεία των παραπάνω

Vasilis Stergioulis

- Embedding Neural Network
- Word2Vec (finalized 2nd Implementation)
- Book Word2Vec Classification
- SVM's
- Θεωρεία των παραπάνω

0.9 Bibliography

1 Sergios Theodoridis , Konstantinos Koutroumbas , Pattern Recognition , fourth edition ,pg130-145

2 Sergios Theodoridis , Konstantinos Koutroumbas , Pattern Recognition , fourth edition ,pg343-352

3 Daniel Jurafksy , Speech and Language Processing , third edition ,pg119-129

4 S.S. Weng et al. , Feature-based recommendations for one-to-one marketing (2004)

5 Konstantinos Diamantaras ,Τεχνητά Νευρωνικά Δίκτυα , pg109-114

6 Minh Dinh ,Vu L. Bui , Doanh C.Bui ,Duong Phi Long, Nguyen D. Vo , Khang Nguyen ,Performance Evaluation of Optimizers for Deformable-DETR in Natural Disaster Damage Assessment