Ανάπτυξη Λογισμικού για Πληροφοριακά Συστήματα 2020-21

Γρηγόρης Καλλίνικος - 1115201500056
Θεοδόσης Παιδάκης - 1115201500118

Part 2

Οδηγίες εκτέλεσης:

- $ *make* : For complete instructions.
- $ *make b|uild*: Compiles the project.
- $ *make t|est*: Runs the unit tests.
- $ *./build/main -x [dataset X folder] -w [dataset W file.csv]*: to run the executable.

Δομές / Σχόλια / Παραδοχές:

— Λόγω της ιδιοτροπίας της δομής των κλικών που δημιουργήσαμε στο πρώτος μέρος του Πρότζεκτ, έπρεπε να σκεφτούμε το πώς θα βρίσκουμε και αποθηκεύουμε τα **προϊόντα που δεν ταιριάζουν** (miss-matches). Κάθε ένα από τα προϊόντα μιας κλίκας λοιπόν, έχει έναν pointer στον ίδιο vector, ο οποίος περιλαμβάνει όλα τα προϊόντα των άλλων κλικών με τα οποία δεν ταιριάζουν.
Κάνοντας κατάλληλο merge miss-match vectors όταν βρίσκουμε νέα ζευγάρια ή μη ζευγάρια, και μαρκάροντας τα κατάλληλα όταν τα έχουμε εμφανίσει, μπορούμε να εξάγουμε τις πιθανές σχέσεις όλων των προιόντων του Dataset.

— Για την αποθήκευση πληροφοριών για το machine learning κομμάτι, **δημιουργήθηκε μια δομή Vector**, στην οποία μπορούμε να αποθηκέυουμε ό,τι δομή χρειαζόμαστε.

— Αρχικά, δημιουργήθηκε ολόκληρη η διαδικασία του **bag of words**, αλλά λόγω του όγκου δεδομένων, δεν φτάνει η ram για την ολοκλήρωση της, οπότε δεν έχει νόημα. Υπάρχει σαν κώδικας όμως.

— Έπειτα, δημιουργούμε τον **idf Vector**, ο οποίος περιέχει όλες τις διαφορετικές λέξεις που υπάρχουν σε όλα τα Specs. Σε αυτόν αποθηκεύουμε την ίδια την λέξη και το idf value της.

— Περνάμε τώρα, ένα ένα τα Specs και υπολογίζουμε την **tf τιμή** της κάθε λέξης που περιέχεται σε καθένα από αυτά.

— Όταν τελειώσει αυτή η διαδικασία, υπολογίζουμε για την κάθε λέξη του idf vector, τον μέσο όρο της tf τιμής της σε όλα τα κείμενα, και σε συνδυασμό με την idf τιμή της, **εξάγουμε τις 3000 λέξεις με το μεγαλύτερο tf-idf value**.

— Σαν παρατήρηση, η τελευταία διαδικασία “ξεσκαρταρίσματος”, διέκρινε τις περισσότερες “νορμάλ” και καθημερινές λέξεις. Οπότε, την θεωρούμε αρκετά επιτυχημένη και βοηθητική.

— Στην συνέχεια, παίρνουμε τον tf-idf vector με τις 3000 λέξεις, και χρησιμοποιώντας το 60% του dataset W, κάνουμε **train το μοντέλο μας** για να βρούμε τους κατάλληλους συντελεστές παλινδρόμησης (βάρη).
Πειραματιστήκαμε με κάποιες τιμές μέχρι να δημιουργήσουμε το κατάλληλο αποτέλεσμα.

— Τέλος, χρησιμοποιούμε το επόμενο 20% του dataset W, αλλά και όλα τα matches και miss-matches που υπολογίσαμε σε προηγούμενα στάδια με την βοήθεια των κλικών, για να κάνουμε **evaluate** το μοντέλο μας.
Έχουμε καταφέρει ένα evaluation score περίπου 92% (success rate).

— Όλες οι απαραίτητες **έξοδοι** εμφανίζονται στην κονσόλα και αποθηκεύονται αντίστοιχα στα αρχεία *matches.txt*, *miss-matches.txt* και *model_validation.txt*.

— Έχουν δημιουργηθεί αρκετά Unit tests που θεωρήσαμε χρήσιμα και γίνεται το απαραίτητο de-allocation στις δομές.

— Το πρόγραμμα χρησιμοποιεί περίπου 6 gb ram συνολικά και τρέχει σε 6-7 λεπτά σε Intel i7 6700k 4.5ghz oc.