

**ΕΡΓΑΣΙΑ**  
**στο μάθημα «Προηγμένα Θέματα Ανάλυσης δεδομένων» ΤΜΗΜΑ**  
**ΨΗΦΙΑΚΩΝ ΣΥΣΤΗΜΑΤΩΝ ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΩΣ ΑΚΑΔ.**  
**ΕΤΟΣ 2024-2025**

**ΤΙΤΛΟΣ ΕΡΓΑΣΙΑΣ**

**ΑΝΑΓΝΩΡΙΣΗ ΣΥΝΑΙΣΘΗΜΑΤΟΣ ΑΠΟ ΚΕΙΜΕΝΟ**

**ΟΝΟΜΑ: ΧΑΚΙ**

**ΕΠΩΝΥΜΟ: ΟΡΟΥΤΣΗ**

**ΑΜ: Ε19239**

## **Εισαγωγή**

Η εργασία αφορά την ανάπτυξη ενός συστήματος το οποίο αναγνωρίζει το συναίσθημα του χρήστη με βάση την κριτική που δίνεται για μία ταινία. Βασικό κομμάτι στην υλοποίηση του συστήματος είναι η εκπαίδευση δύο μοντέλων ανάλυσης συναισθήματος NN(LSTM) και SVM για τα οποία γίνεται σύγκριση. Για την αλληλεπίδραση με τον χρήστη γίνεται ανάπτυξη ενός web app με την χρήση Flask όπου μέσω αυτού γίνεται η ταξινόμηση νέων κριτικών σε πραγματικό χρόνο.

## **Ορισμός προβλήματος**

Τα κύρια προβλήματα για την υλοποίηση της εργασίας είναι η προετοιμασία των δεδομένων για την εκπαίδευση των μοντέλων, η ανάπτυξη των μοντέλων και η επιλογή κατάλληλων μεταβλητών, η σύγκριση των μοντέλων, και η ταξινόμηση νέων κριτικών.

## Παρουσίαση προσέγγισης/ μοντέλου

### NN(LSTM) (αρχείο LSTM\_model.py)

#### **ΒΗΜΑ 1ο**

Γίνεται η εισαγωγή του αρχείου δεδομένων και στην συνέχεια η 'εξερεύνηση' του με την χρήση των κατάλληλων εντολών .

#### **ΒΗΜΑ 2ο**

Αφού διαπιστώθηκε ότι το αρχείο περιέχει επιπλέον στήλες από αυτές που χρειάζονται για την υλοποίηση της εργασίας αφαιρέθηκαν οι αχρείαστες στήλες και χρησιμοποιήθηκαν μόνο οι στήλες 'review' και 'label' .

#### **ΒΗΜΑ 3ο**

Στην στήλη 'label' περιέχονται τρεις κατηγορίες για τα reviews 'pos', 'neg', 'unsup'. Για την εκπαίδευση και την δοκιμή των μοντελών γίνεται χρήση μόνο των κατηγοριών 'pos', 'neg', τα οποία αντιστοιχούν σε 25.000 reviews το καθένα και η κατηγορία 'unsup' αντιστοιχεί σε 50.000 reviews. Οπότε, μετά την αφαίρεση των reviews που αντιστοιχούν στην κατηγορία 'unsup' το σύνολο των reviews που θα χρησιμοποιηθούν είναι 50.000. Δηλαδή, το αρχείο δεδομένων μειώθηκε κατά 50%.

#### **ΒΗΜΑ 4ο**

Οι κατηγορίες 'pos' και 'neg' της στήλης 'label' μετατράπηκαν σε αριθμητικές '1' και '0' το οποίο βοηθάει πολύ στις δυαδικές ταξινομήσεις.

## **ΒΗΜΑ 5ο**

Με την χρήση της συνάρτησης preprocess γίνεται ο καθαρισμός των reviews στα οποία αφαιρούνται τα html tags και το περιεχόμενό τους, μετατρέπονται όλες οι λέξεις σε πεζά και χωρίζονται σε μεμονωμένες λέξεις, αφαιρούνται όλοι οι μη αλφαριθμητικοί χαρακτήρες, αφαιρούνται τα stopwords, οι λέξεις επανέρχονται στην αρχική μορφή τους με την χρήση του stemmer, και επιστρέφονται οι λέξεις με κενό ανάμεσά τους.

## **ΒΗΜΑ 6ο**

Ανανεώνεται η στήλη 'review' με τα νέα "καθαρισμένα" reviews.

## **ΒΗΜΑ 7ο**

Χωρίζονται τυχαία τα δεδομένα σε train και test sets σε 80% και 20% αντίστοιχα

## **ΒΗΜΑ 8ο**

Αρχή της διαδικασίας για την προετοιμασία των reviews για την χρήση τους σε μοντέλο νευρωνικών δικτύων(NN).

## **ΒΗΜΑ 9ο**

Εκπαίδευση tokenizer πάνω στο X\_train set για την μετατροπή των κειμένων από τα reviews σε αριθμητικές αναπαραστάσεις.

## **ΒΗΜΑ 10ο**

Υπολογισμός μοναδικών λέξεων λεξιλογίου (62.873 λέξεις).

## **ΒΗΜΑ 11ο**

Δημιουργία ακολουθιών και έπειτα επεξεργασία του μήκους των ακολουθιών.

## **ΒΗΜΑ 12ο**

Αρχή δημιουργίας word embeddings με τη χρήση word2vec.

## **ΒΗΜΑ 13ο**

Εκπαίδευση μοντέλου word2vec πάνω στα reviews όπου έχει εφαρμοστεί split()

## **ΒΗΜΑ 14ο**

Γίνεται αποθήκευση του μοντέλου word2vec και έπειτα φορτώνεται.

## **ΒΗΜΑ 15ο**

Δημιουργία embedding\_matrix\_vocab.

## **ΒΗΜΑ 16ο**

Κατασκευή LSTM μοντέλου το οποίο έχει είσοδο max\_words και περιέχει ένα embedding layer με τις τιμές του embedding\_matrix\_vocab, ένα LSTM layer με 128 νευρωνες, και ένα τελικό dense layer με sigmoid activation εφόσον πρέπει να γίνει δυαδική ταξινόμηση.

## **ΒΗΜΑ 17ο**

Γίνεται ορισμός παραμέτρων για την εκπαίδευση του μοντέλου και στην συνέχεια γίνεται η εκπαίδευση του για 6 epochs χρησιμοποιώντας adam optimizer και binary\_crossentropy loss και υπολογίζεται ο χρόνος της εκπαίδευσης.

## **ΒΗΜΑ 18ο**

Μετατροπή των προβλέψεων σε κατηγορίες '1' και '0' , ώστε να γίνει η αξιολόγηση του μοντέλου σε σχέση με το Y\_test set.

#### **ΒΗΜΑ 19ο**

Γίνεται αποθήκευση του LSTM μοντέλου και του tokenizer, ώστε να μπορούν να χρησιμοποιηθούν στην διαδικτυακή εφαρμογή.

#### **ΒΗΜΑ 20ο**

Δημιουργία συνάρτησης LSTMpredictionTime για τον υπολογισμό του χρόνου ταξινόμησης ώστε να γίνει ο υπολογισμός του μέσου χρόνου ταξινόμησης σε 20 τυχαία reviews.

#### **ΒΗΜΑ 21ο**

Γίνεται αξιολόγηση της απόδοσης του μοντέλου με μετρικές όπως precision, recall, f1, accuracy.

### **SVM (αρχείο SVM\_model.py)**

Για την δημιουργία του SVM μοντέλου επαναλαμβάνονται όλα τα βήματα από το ΒΗΜΑ 1ο έως και το ΒΗΜΑ 7ο από τα βήματα για την δημιουργία του LSTM.

#### **ΒΗΜΑ1ο – 7ο**

(Βλέπε βήματα από lstm).

#### **ΒΗΜΑ 8ο**

Αρχή της διαδικασίας για την προετοιμασία των reviews για την χρήση τους σε SVM μοντέλο.

#### **ΒΗΜΑ 9ο**

Εκπαίδευση vectorizer πάνω στο X\_train set για την μετατροπή των κειμένων από τα reviews σε αριθμητικές αναπαραστάσεις.

#### **ΒΗΜΑ 10ο**

Δημιουργία SVM μοντέλου με χρήση linear kernel και C=1.

#### **ΒΗΜΑ 11ο**

Εκπαίδευση μοντέλου SVM και υπολογισμός χρόνου εκπαίδευσης

#### **ΒΗΜΑ 12ο**

Γίνεται αποθήκευση του SVM μοντέλου και του vectorizer, ώστε να μπορούν να χρησιμοποιηθούν στην διαδικτυακή εφαρμογή.

### **ΒΗΜΑ 13ο**

Δημιουργία συνάρτησης SVMpredictionTime για τον υπολογισμό του χρόνου ταξινόμησης ώστε να γίνει ο υπολογισμός του μέσου χρόνου ταξινόμησης σε 20 τυχαία reviews.

### **ΒΗΜΑ 14ο**

Γίνεται αξιολόγηση της απόδοσης του μοντέλου σε μετρικές όπως precision, recall, f1, accuracy.



## **FLASK APP (αρχεία webApp.py, home.htmlm, home\_css.css)**

### **BHMA 1ο**

Γίνεται φόρτωση του SVM μοντέλου και του vectorizer.

### **BHMA 2ο**

Δημιουργά της συνάρτησης preprocess για να γίνει ο καθαρισμός του νέου review στο οποίο αφαιρούνται τα html tags και το περιεχόμενό τους, μετατρέπονται όλες οι λέξεις σε πεζά και χωρίζονται σε μεμονωμένες λέξεις, αφαιρούνται όλο τα μη αλφαριθμητικά, αφαιρούνται τα stopwords, οι λέξεις επανέρχονται στην αρχική μορφή τους με την χρήση του stemmer, και επιστρέφονται οι λέξεις με κενό ανάμεσά τους.

### **BHMA 3ο**

Δημιουργία συνάρτησης prediction για την ταξινόμηση του νέου review το οποίο “καθαρίζεται” με την εφαρμογή της συνάρτησης preprocess και μετατρέπεται σε αριθμητική αναπαράσταση και στην συνέχεια γίνεται η ταξινόμηση με την χρήση του SVM μοντέλου.

### **BHMA 4ο**

Δημιουργία web app με την χρήση της βιβλιοθήκης flask το οποίο περιέχει:

- Την αρχική σελίδα home.html.
- Το αποτέλεσμα της ταξινόμησης το οποίο αποστέλλεται στο home.html μαζί με την ανάλογη εικόνα. Στο αρχείο home.html υπάρχει η συνθήκη όπου ελέγχεται αν υπάρχει η αποστολή από το αποτέλεσμα της ταξινόμησης ώστε να εμφανιστούν στην σελίδα η ανάλογη ταξινόμηση και εικόνα.

## **Πειραματική μελέτη**

### **NN (LSTM) μοντέλο :**

Precision: 0.86

Recall: 0.90

F1: 0.88

Accuracy: 0.88

Χρόνος εκπαίδευσης : 155.81 secs

Μέσος χρόνος ταξινόμησης σε 20 τυχαία reviews : 0.066 secs

### **SVM μοντέλο :**

Precision: 0.89

Recall: 0.91

F1: 0.90

Accuracy: 0.90

Χρόνος εκπαίδευσης : 879.72 secs

Μέσος χρόνος ταξινόμησης σε 20 τυχαία reviews : 0.015 secs

### **Σύγκριση :**

- Μετρικές  
Το SVM μοντέλο είναι ελάχιστα πιο καλό από το LSTM μοντέλο σε όλες τις μετρικές.
- Χρόνος εκπαίδευσης  
Παρατηρείται μεγάλη διαφορά στον χρόνο εκπαίδευσης των μοντέλων καθώς το SVM απαιτεί τουλάχιστον 5 φορές τον χρόνο εκπαίδευσης του LSTM.
- Χρόνος ταξινόμησης  
Το LSTM μοντέλο απαιτεί αρκετά περισσότερο χρόνο από το SVM μοντέλο για την ταξινόμηση ενός νέου review.

### **Συμπεράσματα**

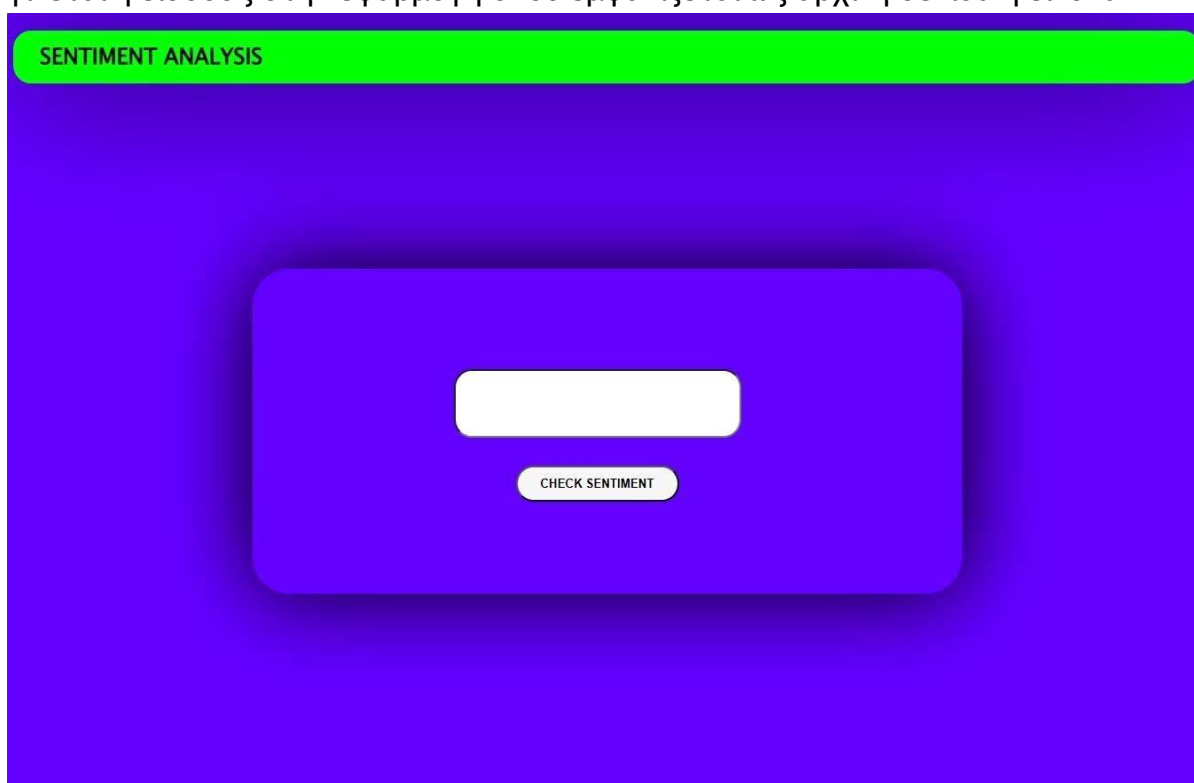
Με βάση τον χρόνο εκπαίδευσης του κάθε μοντέλου συμπεραίνουμε πως ένα μοντέλο NN μπορεί να εκπαιδευτεί πολύ πιο γρήγορα σε μεγάλα αρχεία δεδομένων και να έχει ισάξια αποτελέσματα στις συγκρίσεις των διάφορων μετρικών. Οπότε, όσο μεγαλύτερο είναι το αρχείο δεδομένων όπου θα γίνει η εκπαίδευση του μοντέλου, το NN(LSTM) θα αποκτά όλο και μεγαλύτερο προβάδισμα.

Ωστόσο, στην περίπτωση μας, η χρήση των μοντέλων πρόκειται να γίνει για ταξινόμηση νέων κριτικών σε πραγματικό χρόνο. Για αυτό και καταλήγουμε στο συμπέρασμα πως το SVM μοντέλο είναι αρκετά πιο ευνοϊκό ως προς την χρήση του στην εφαρμογή μας η οποία πρόκειται για ταξινομήσεις νέων κριτικών σε πραγματικό χρόνο, εφόσον ο χρόνος που χρειάζεται το SVM μοντέλο είναι

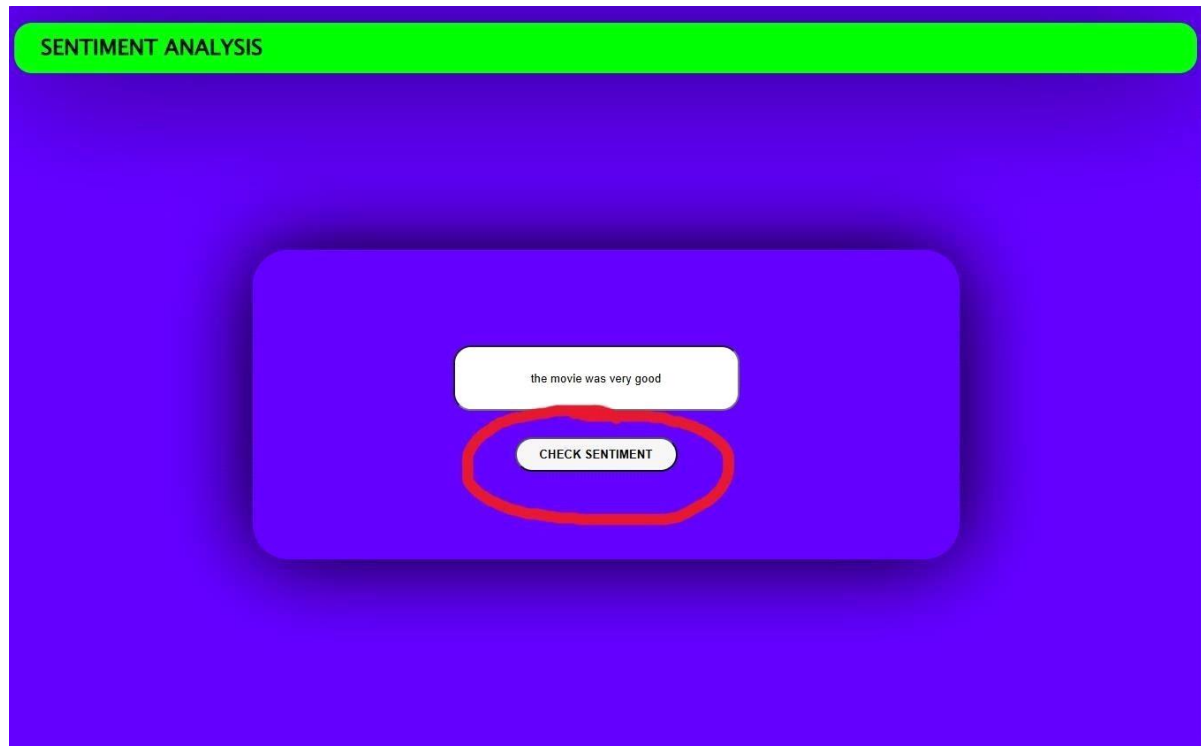
τουλάχιστον 4 φορές μικρότερος από τον χρόνο που χρειάζεται το LSTM μοντέλο για μια νέα ταξινόμηση.

## ΟΔΗΓΙΕΣ ΧΡΗΣΗΣ ΕΦΑΡΜΟΓΗΣ

Πληκτρολογώντας στο terminal την εντολή `python webApp.py` θα ενεργοποιηθεί η πύλη για την διαδικτυακή εφαρμογή και στην συνέχεια με την διεύθυνση: <http://127.0.0.1:8000> γίνεται η είσοδος στην εφαρμογή όπου εμφανίζεται ως αρχική σελίδα η εικόνα :



Ο χρήστης συμπληρώνει την κριτική του στο λευκό πεδίο και στην συνέχεια πατάει το κουμπί “CHECK SENTIMENT” για να δει τα αποτελέσματα της κριτικής του.




The image shows a web interface for sentiment analysis. At the top, there is a green header bar with the text "SENTIMENT ANALYSIS" in white. Below this, the main area has a dark blue background. In the center, there is a white rounded rectangle containing a text input field and a button. The text input field contains the text "the movie was very good". Below the input field, the button "CHECK SENTIMENT" is highlighted with a red hand-drawn oval.

Αν η κριτική του χρήστη είναι θετική τότε θα εμφανιστεί η ανάλογη εικόνα και το συναίσθημα και δίπλα εμφανίζεται το πεδίο για να εισάγει μια νέα κριτική:

SENTIMENT ANALYSIS

CHECK SENTIMENT

Sentiment: positive




Αν η κριτική του χρήστη είναι αρνητική τότε θα εμφανιστεί η ανάλογη εικόνα και το συναίσθημα και δίπλα εμφανίζεται το πεδίο για να εισάγει μια νέα κριτική:

SENTIMENT ANALYSIS

CHECK SENTIMENT

Sentiment: negative



## **BIBΛΙΟΓΡΑΦΙΑ**

[Introduction to Recurrent Neural Networks - GeeksforGeeks](#)

[Sentiment Analysis with an Recurrent Neural Networks \(RNN\) - GeeksforGeeks](#)

[Word Embedding using Word2Vec - GeeksforGeeks](#)

[Pre-trained Word embedding using Glove in NLP models - GeeksforGeeks](#)

[Sentiment Analysis using Recurrent Neural Network\(RNN\),Long Short Term Memory\(LSTM\) and Convolutional Neural Network\(CNN\) with Keras. | by Muhammad Luay | Medium](#)

[Sentiment Analysis using SVM. Sentiment Analysis is the NLP technique... | by Vasista Reddy | ScrapeHero | Medium](#)

[Sentiment Analysis — using LSTM & GloVe Embeddings | by Skillcate AI | Medium](#)

[Building and Deploying a Sentiment Analysis Model Using LSTM and Pre-trained GloVe Embeddings | by SIRI BATCHU | Medium](#)

[Sentiment Analysis using LSTM model & Flask web app | Flask Code Part - 1 - YouTube](#)

[Sentiment Analysis Web App Using Python and Flask CoderPros](#)

[Part 6: Machine Learning Model Deployment with Flask for Sentiment Analysis Project](#)

[Sentiment Analysis with LSTM | Deep Learning with Keras | Neural Networks | Project#8](#)

[IMDB Sentiment Analysis Using SimpleRNN Layer From Tensorflow | Working With Sequential Data \(Text\)](#)



[Project 25: IMDB Movie Review Sentiment Analysis Using Deep Learning](#)  
[DL Project 10. Sentiment Analysis on IMDB Reviews with LSTM | Deep Learning Projects](#)

[Machine Learning on Movie Reviews \(IMDB Dataset - 50k reviews\) | Machine Learning Project 11](#)