

# Deep Learning for NLP

Student name: *Dimitrios Rammos*  
sdi: *sdi1900161*

---

Course: *Artificial Intelligence II (M138, M226, M262, M325)*  
Semester: *Fall Semester 2023*

---

## Contents

<b>1</b>	<b>Abstract</b>	<b>2</b>
<b>2</b>	<b>Data processing and analysis</b>	<b>2</b>
2.1	Pre-processing . . . . .	2
2.2	Analysis . . . . .	2
2.3	Data partitioning for train, test and validation . . . . .	3
2.4	Vectorization . . . . .	3
<b>3</b>	<b>Algorithms and Experiments</b>	<b>3</b>
3.1	Experiments . . . . .	3
3.2	Experiments . . . . .	3
3.2.1	Table of trials . . . . .	3
3.3	Hyper-parameter tuning . . . . .	3
3.4	Optimization techniques . . . . .	4
3.5	Evaluation . . . . .	4
3.5.1	ROC curve . . . . .	4
3.5.2	Learning Curve . . . . .	4
3.5.3	Confusion matrix . . . . .	4
<b>4</b>	<b>Results and Overall Analysis</b>	<b>5</b>
4.1	Results Analysis . . . . .	5
<b>5</b>	<b>Bibliography</b>	<b>5</b>

## 1. Abstract

<Briefly describe what's the task and how you will tackle it.>

## 2. Data processing and analysis

### 2.1. Pre-processing

Let's describe the preprocessing. First off all I have separated my code in cells. I do pre processing for 3 csv file( train, valid and test).

I read the file and I change the 'Text' column for dataset removing hashtags, at(@) etc.

1. First I delete the hashtags with the word. For example, if my string is "Hello #november" the result will be "Hello", I remove "#november".
2. After, I remove all urls from my strings texts in dataset, for example the "Hello https://www.youtube.com" will beacame "Hello".
3. Subsequently, I remove all numbers as words. For example, the "Hello 123 Jim" will become "Hello Jim".
4. I remove all stopwords, greek stopwords.
5. I remove the accents from the words.
6. Because some word may have been not accent from the beginning, I create a new table with not accents and I remove again the stopwords from dataset.
7. I remove the @words(at-words). For example the string "Hello @jim hello" will beacame "Hello hello".
8. I remove all punctuatuions from dataset.
9. In the end I do tokenizer for my strings. For example, the "hello jim goodmorn-ing" will became ['hello', 'jim', 'became'].

### 2.2. Analysis

I read the files .csv from input and I create the dataset for train, valid and test. I do the pre-processing for evrey dataset and I save it. In my code from kaggle, I have functions from Spacy library. I try do pre-process with spacy functions but I noticed that the preprocessing was slowly. In nootebook you can see this functions but I not

used.

At first I did lemmatization but I saw that my score instead of increasing towards the positive was worse and slower so I deleted it.

### 2.3. Data partitioning for train, test and validation

The files I used were from the work compention Project - 1.

### 2.4. Vectorization

For my project, I used FFNN on train, validation and test datasets. I create a new model for it and I use FFNN model algorithm.

## 3. Algorithms and Experiments

### 3.1. Experiments

### 3.2. Experiments

(THIS IS TEST-STEPS FROM PREVIOUS EXERCISE)

The score was very close to 0.386. When for example I removed single letters like "s" or anything else the score decreased instead of improving. That's why I haven't implemented any functionality in my pre-process yet.

At first I tried everything with the spacy library as it seemed easier to delete words or symbols but I noticed that it took too long for each process and so I changed the method and tried other ways to do the process.

Trial				Score

Table 1: Trials

#### 3.2.1. Table of trials.

### 3.3. Hyper-parameter tuning

You can see the code for hyper-parameter tuning in my notebook. The result is: Best params: hiddenSize = 151, learningRate = 5.717760419313267e-05

Best params: {'hidden\_size': 151, 'learning\_rate': 5.717768419313267e-05}

Figure 1: Hyper Parameter tuning

### 3.4. Optimization techniques

I try change the preprocess for a better results. For example, lemmatization, emojis, english stopwords, not accents stopwords.

### 3.5. Evaluation

#### 3.5.1. ROC curve. The ROC curve.

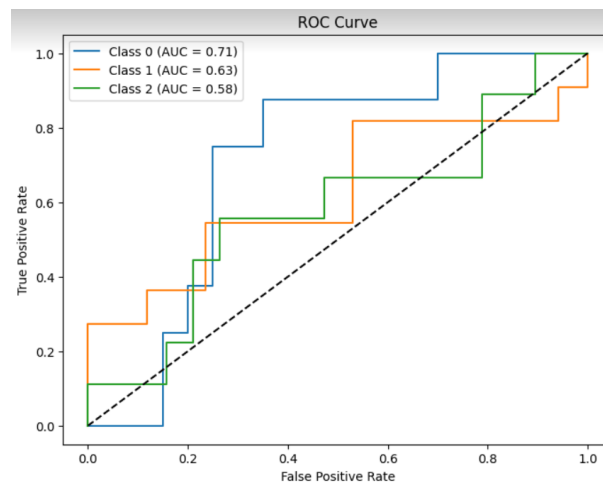


Figure 2: Roc curve

#### 3.5.2. Learning Curve. The learning curve.

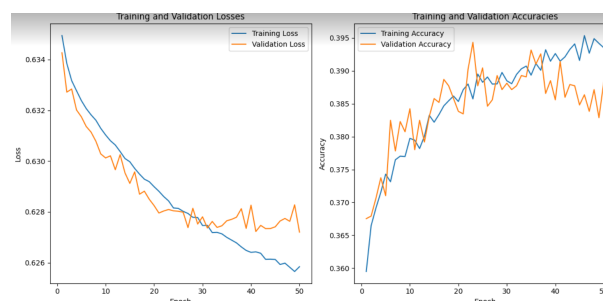


Figure 3: Learning curve

#### 3.5.3. Confusion matrix. The confusion matrix

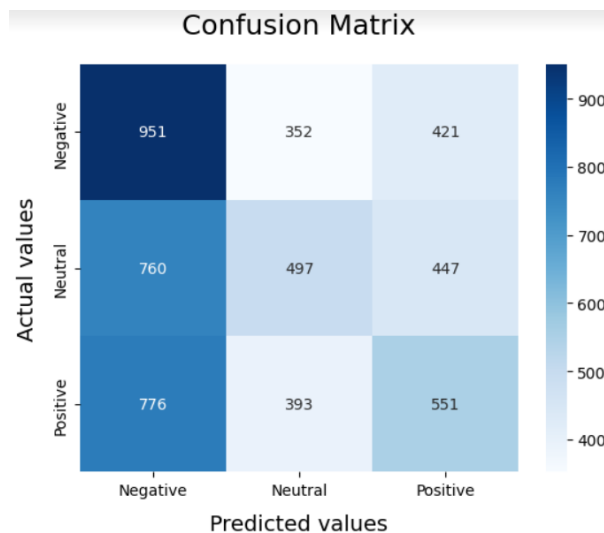


Figure 4: Confusion matrix

## 4. Results and Overall Analysis

### 4.1. Results Analysis

I don't think it's a good performance and I could definitely have a better performance. Maybe it's due to the pre process. I think some corrections could be made there.

However, the dataset may also create some problems and difficulties in the code. Surely other corrections could be made, such as words without meaning in English or Greek. Sticky words from syntactical errors of the users who wrote the texts and we could separate them. Spelling mistakes that could be fixed. Make sure to do lemmatization without spoiling the score and definitely other corrections that I haven't considered yet.

## 5. Bibliography