Artificial Intelligence II

Homework 1

Eirini Kolimatsi

Student ID: 7115112200015
Email: cs22200015@di.uoa.gr

MSc Computer Science
University of Athens
Academic Year: 2022-2023

Contents

1 Data Preparation

Libraries Firstly, all required libraries should be loaded. For the needs of this assignments modules from scikit-learn (sklearn), nltk, pandas, numpy, re, matplotlib, contractions, google.colab and wordcloud.

Data Loading In order to access the data provided for the assignment, the csv file was uploaded in Google drive. Then the notebook got "connected" to Google Drive, to allow reading the file from a path. The path that leads to the file is /content/drive/path/to/file, where path/to/file should be replaced with the actual location in Google Drive where the file was uploaded.

1.1 Data Cleaning

The provided data have 3 columns: rating, URL and review not needed. Reviews are expressed in a scale from 1 to 10. As a result, all negative reviews (with a score below or equal to 4) are marked with 0 and all positive reviews (with a score above or equal to 7) are marked with 1.
Regarding the URL, we can separate the movie identifier from the rest of the URL. This feature cannot be used per se in the training, but someone may decide to use it to optimize the way data are split in the train and validation sets.
Going through the reviews, it is clear that the data need to be cleaned. Firstly, reviews include HTML tags (eg <br>) which cannot help understand the sentiment of the review and need to be removed. Similarly for punctuation points. Also, the characters should all be in lower case to avoid the same word being interpreted as a different one given the case. Lastly, given that reviews are everyday simple text there are contractions present in the text. For example, isn't is used instead of is not. To "expand" these abbreviated words, a dedicated library is used, that is called contractions.

On top of these steps, there are many more linguistic cleaning steps that can be applied, of which three have been selected to be examined. Firstly, the removal of stopwords, stemming and lemmatization.

What is suggested is to explore which of the aforementioned techniques of text cleaning better serves the given dataset and the purposes of model training to choose which ones to apply. Nevertheless, some of the cleaning steps ought to be done in all cases, eg HTML tags removal and lower case adaptation. To decide which methods to use, a quick model training will be performed to find out if it helps out the model or not.

1.2 Data Exploration

Before training the model, it is useful to explore further the dataset. We observe that the number of positive and negative reviews is very similar, so the dataset is balanced. Also, there are no missing values and as a result we should not remove them or fix them in another way.
To gain a deeper understanding of the dataset, wordcloud plots are generated for both positive and negative reviews. The positive wordcloud include words like great, love etc.

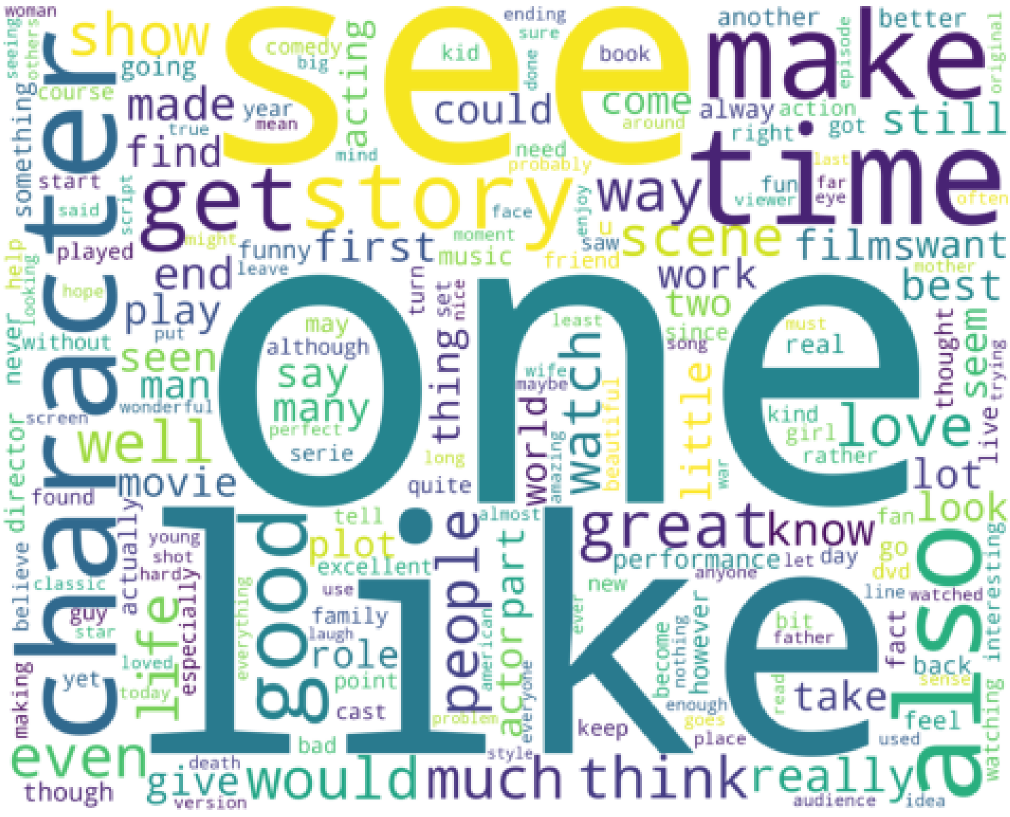

Figure 1: Positive Wordcloud

The negative wordcloud contains words like bad and worst.

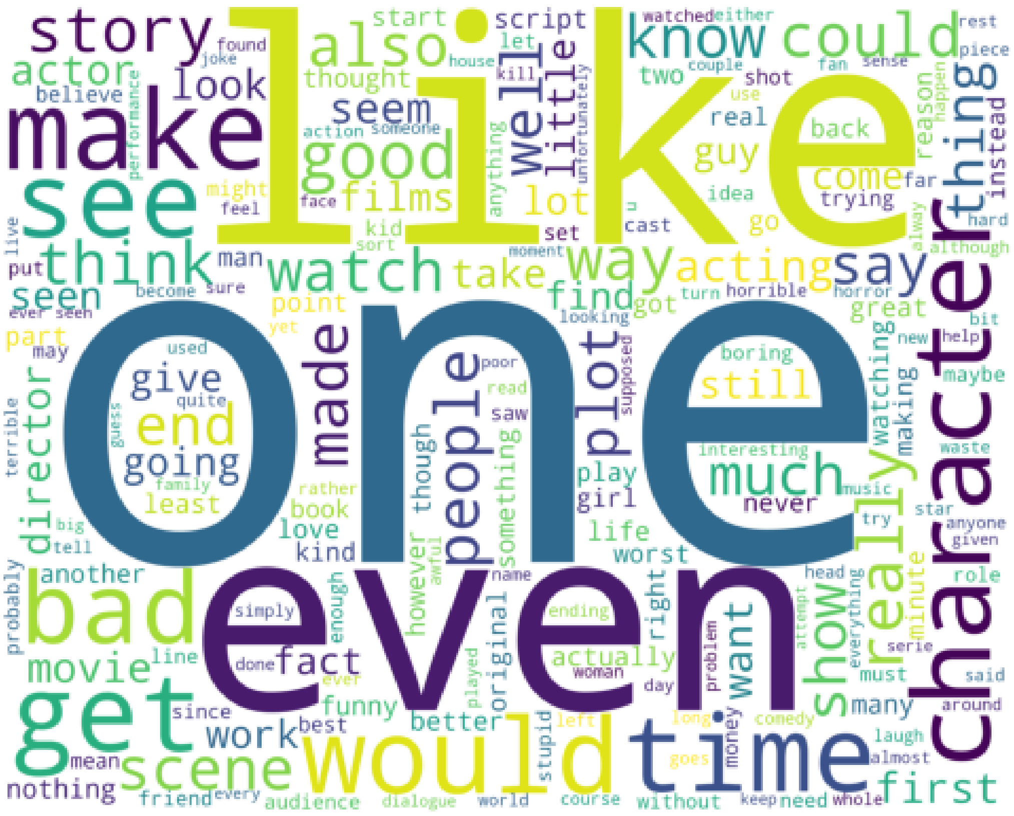

Figure 2: Negative Wordcloud

We might notice that the most frequent words from both wordclouds are similar, which leads to the assumption that words that are often mentioned might not be the most suitable to extract sentiment.

2 Training

For the training, the approach used is to try to build the features used, as well as the vectorisation methods from a simple selection to more complicated combinations and along the way evaluate the performance. At the end, the most suitable model is selected to be fine tuned and examined more for its performance.

Regarding the train test split, given that the dataset is balanced a 80-20 split is chosen.

The first attempt for model training was done only doing the necessary data cleaning, as described in the Data Cleaning section and with the help of a CountVectorizer to interpret the reviews. Two different values where given for C, 0.01 and 1. The model seems to perform at least fine given the preliminary results.
Precision (train): 0.8904639877869255
Precision (validation): 0.8744560487380331
Recall (train): 0.9072325297189201
Recall (validation): 0.8919218819351975
F1 Score (train): 0.8987700520045127
F1 Score (validation): 0.8831026148099319

Then, for the second attempt instead of the CountVectorizer, TfidfVectorizer is used with different parameters (stopwords removal, min_df and ngram_range). This vectorizer performs worse than the CountVectorizer in the first attempt:
Precision (train): 0.8143405889884763
Precision (validation): 0.8045427375971309
Recall (train): 0.8854485715876816
Recall (validation): 0.8872775214238628
F1 Score (train): 0.848407235473027
F1 Score (validation): 0.8438871473354231

For the third attempt, stopwords removals, contractions removal and lemmatization was applied, as well as both CountrVectorizer and TfidfVectorizer. The CountVectorizer performs better:
Precision (train): 0.8855989966737554
Precision (validation): 0.8688102893890676
Recall (train): 0.9050431875174143
Recall (validation): 0.8882314266929652
F1 Score (train): 0.895215521993165
F1 Score (validation): 0.8784135240572172

For the fourth attempt, stopwords removals, contractions removal and stemming was applied, as well as both CountrVectorizer and TfidfVectorizer. The CountVectorizer performs better:
Precision (train): 0.8835975972725797
Precision (validation): 0.8693957115009746
Recall (train): 0.9050496092234355
Recall (validation): 0.8985896574882472
F1 Score (train): 0.8941949616648412
F1 Score (validation): 0.8837516512549537

Finally, it seems like the 1st and the 4th model perform very close and they're the best approaches out of the 4. However, the 1st training attempt will be selected because it performs marginally better across mode metrics than the 4th.

3 Results

After selecting the model, it is tuned to select the best C and then it is cross validated (10-fold). Finally, the results are: Precision (train): 0.8901395589598431
Precision (validation): 0.8753139473157765
Recall (train): 0.9061074556155203
Recall (validation): 0.8915052273063824
F1 score (train): 0.8980523558387767
F1 score (validation): 0.8833186823944799

The confusion matrix has been generated for both training and validation results.

To run the produced model with test data, at the end of the notebook the path can be specified, and the following cells should be executed.

4 References

Sklearn documentation
Stackoverflow

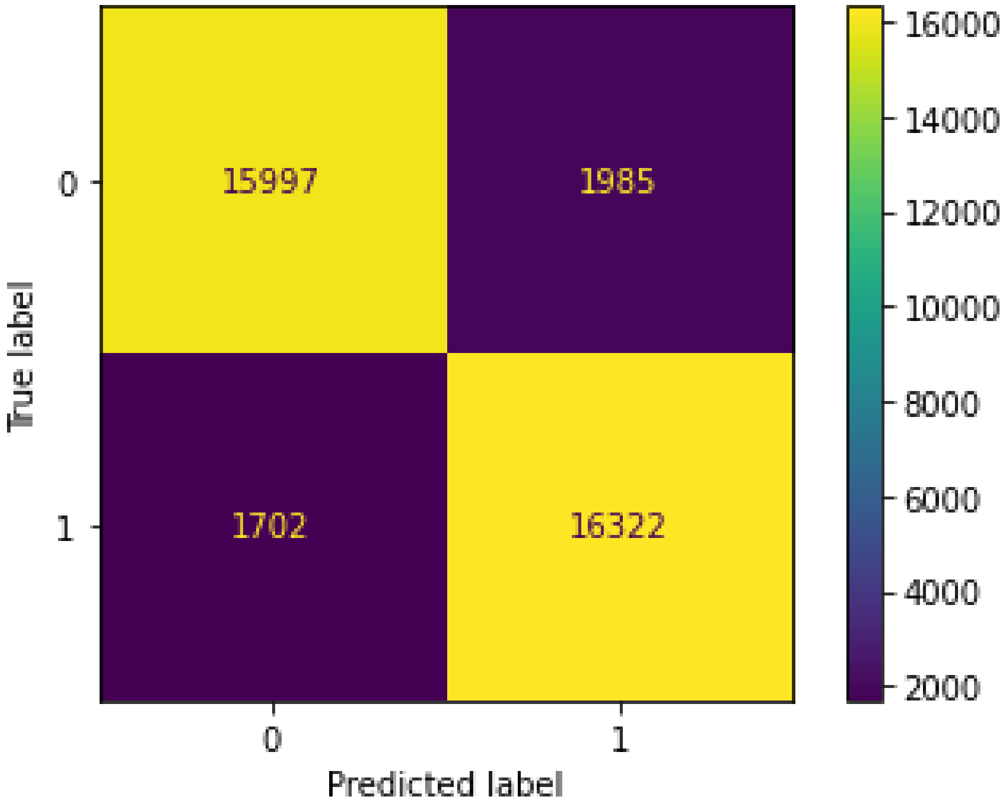

Figure 3: Confusion Matrix (train set)

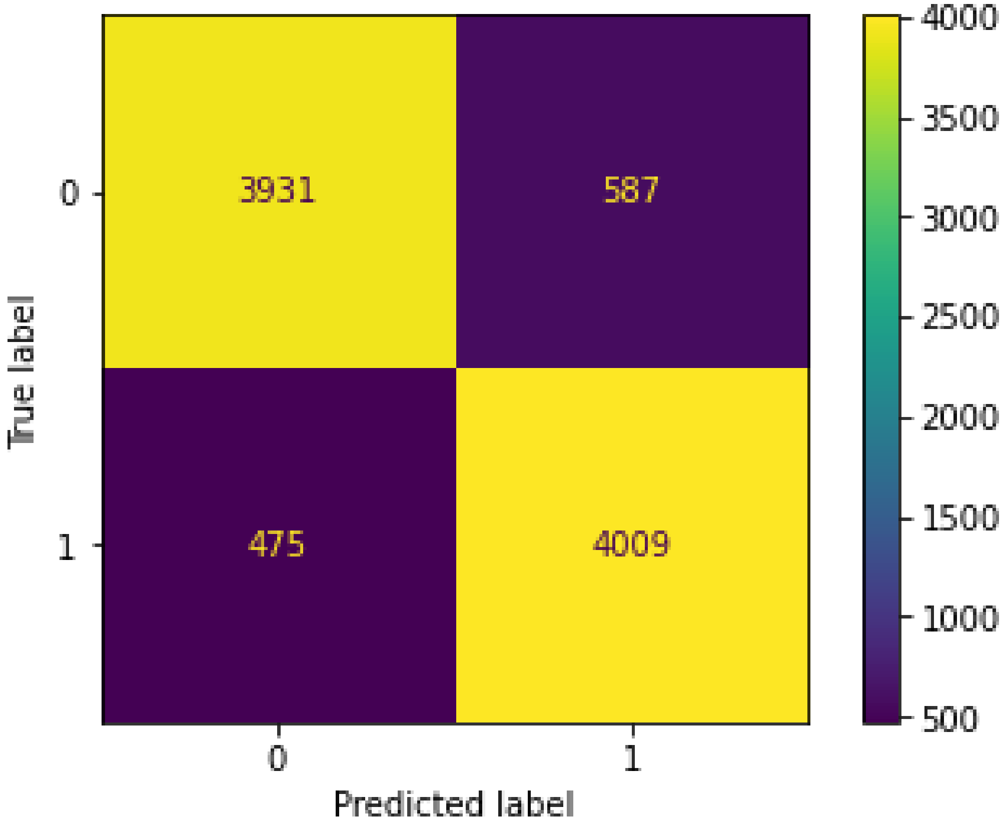

Figure 4: Confusion Matrix (validation set)