Artificial Intelligence II - Homework 1

Ioanna Oikonomou

November 2022

Contents

1 Introduction

In this report I am going to explain my solution to the first assignment which requires developing a sentiment classifier using logistic regression on an imdb dataset that was given.

2 Assignment

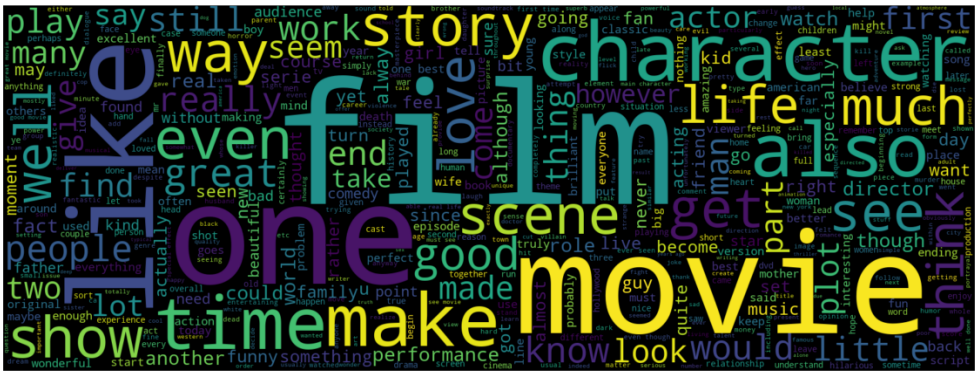
2.1 Data preprocessing

The review of the dataset contained some useless information that could complicate things for my model. That's why I applied a preprocessing function and got rid of duplicate data. Also, I added a sentiment column to the dataset to be able to make the predictions for good and bad movies.

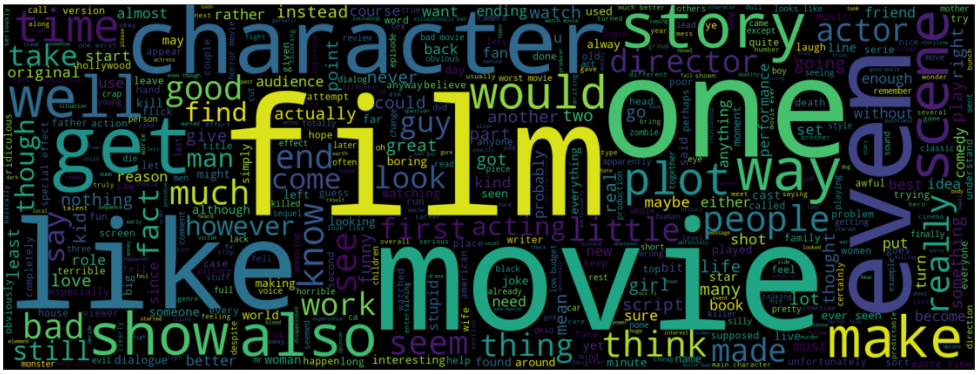
My preprocessing function follows the following steps:
-Transforming all letters to lowercase
-Removing urls
-Removing html line breakers

-Tokenizing the reviews
-Removing punctuation
-Removing stopwords
-Removing numbers
-Lemmatizing the data

Word clouds were used to vidualize the key words and most frequently used words in positive and negative reviews.
Word cloud of positive reviews visualized:



Word cloud of negative reviews visualized:



2.2 Data Transformation

For transforming the data I used TfidfVectorizer in order to avoid using CountVectorizer and TfidfTransformer to normalize the data.

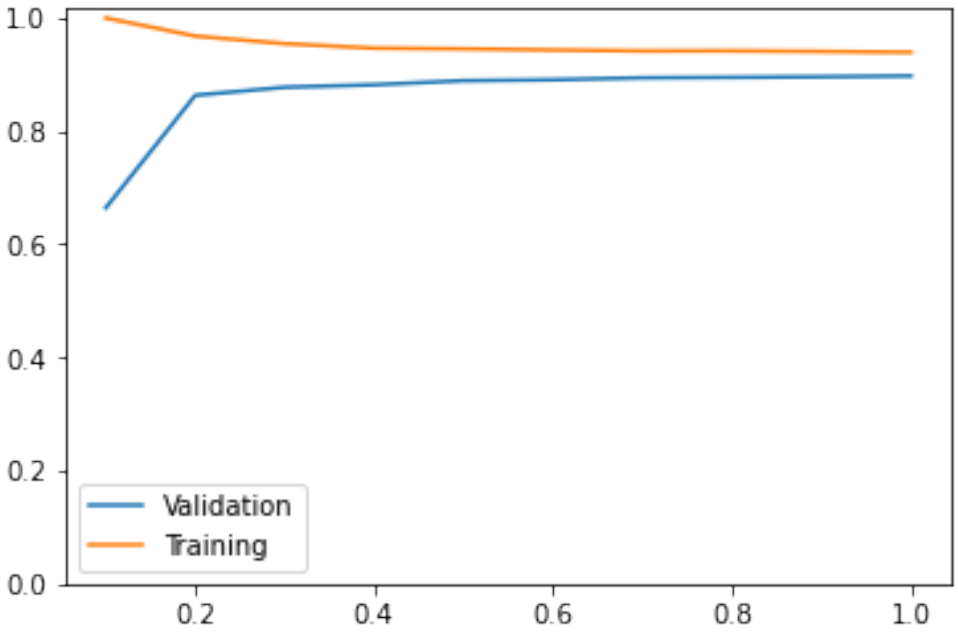
2.3 Hypeparameter tuning

The hyperparameters used in my model were chosen by using grid search. Note that this function takes too much time and I only ran it once to get the best parameters, which were C=10, penalty="l2", solver="newton-cg".

2.4 Learning curve

For the learning curve of my model I modified the code that was shown at the second lecture regarding this assignment. The link of the code

The result of the learning curve was the following:

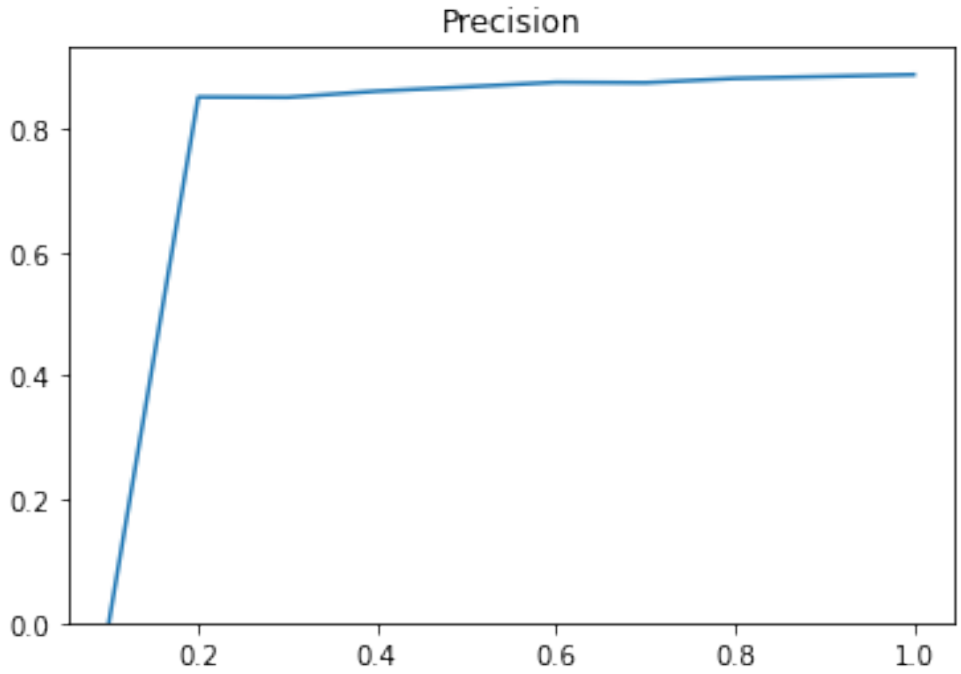


We can see above that for a small sample of the training set, the model overfits but as the sample grows, the training score comes closer to the validation score.

2.5 Evaluation

In order to evaluate the model I got the classification report which contains information about f1score, precision, recall and support.

Below you can see some diagrams that show how precision changes as the sample size grows.



We can see that precision start with low scores but after the sample size reaches the 0.2 of the training size, it gains scores around 0.8.
The digram of the recall scores is similar and it can be seen in my code.

Here, I'd like to stress that the beginning of these graphs, meaning the diagrams in their beginning (for small sample size) depends on the data that this sample contains. The precision and recall could as easily be close to 1 in the beginning and then decrease as the sample size grows.

3 Testing

The last part of my code is a section you can use to test with the test set. I'm not sure this is what you needed in order to test it. I hope it helps :)

4 Sources

-Scikit Learn
-Hyperparameter tuning
-Lecture's slides from eclass
-Stack Overflow