TEXT MINING ASSIGNMENT 1 REPORT
LEIDEN UNIVERSITY FACULTY OF SCIENCE, LIACS
2023-2024

Text Mining Assignment 1

Yağmur DOĞAN, Barbaros IŞIK
October 2023

I. Introduction

THe assignment 1 is an assignment where we performed a text categorization task with benchmark data in scikit-learn, understand the effect of using different types of feature weights and evaluate text classifiers with the suitable evaluation metrics.

In this assignment, we compared three types of features for our classifiers, which are: Naive Bayes, Stochastic Gradient Descent Classifier (SGDC) and Decision Tree Classifier (DTC).

With these three classifiers, as requested from us in the 3rd task of Assignment; We compared the count vectorizer, tf (term frequency), and tf-idf features.

II. Methodology

A. Naive Bayes Classifier

As mentioned in Task 2, the first Classifier we used was the Naive Bayes Classifier. The Naive Bayes Classifier is a machine learning algorithm employed for the task of classification. Its foundation rests upon Bayes' theorem, a mathematical principle that facilitates the estimation of probabilities based on prior knowledge of related events. The term ”naive” in its nomenclature is derived from an assumption of feature independence, a simplifying assumption often made for computational convenience. [7]

B. Stochastic Gradient Descent Classifier

The Stochastic Gradient Descent Classifier (SGDC) is a machine learning algorithm used for training linear classifiers, such as Support Vector Machines (SVM) and logistic regression models. It's an optimization technique that updates the model's parameters in an iterative way to minimize a cost function. SGDC is well-suited for large scale and online learning scenarios.

C. Decision Tree Classifier

Decision Tree Classifier (DTC) are a non-parametric supervised learning method used for classification and regression. Decision trees learn from data to approximate a sine curve with a set of if-then-else decision rules. The deeper the tree, the more complex the decision rules and the fitter the model.

A decision tree classifier is just like a flowchart diagram with the terminal nodes representing classification outputs/decisions. Starting with a dataset, you can measure the entropy to find a way to split the set until all the data belonngs to the same class. [4]

D. Count Vectorization, Term Frequency & Term Frequency-Inverse Document Frequency

Count Vectorization involves counting the number of occurrences each words appears in a document (i.e distinct text such as an article, book, even a paragraph!). Python's Sci-kit learn library has a tool called CountVectorizer to accomplish this.

Term Frequency (TF) refers to how much a term (i.e. a word) appears in a document.

Inverse document frequency (TF-IDF) refers to how common or rare a term appears in a document. [6]

III. Practises

A. Packages & Libraries

```
    from sklearn import metrics
from sklearn.datasets import fetch_20newsgroups
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.feature_extraction.text import TfidfTransformer
from sklearn.naive_bayes import MultinomialNB
from sklearn.linear_model import SGDClassifier
from sklearn.tree import DecisionTreeClassifier
from sklearn.metrics import accuracy_score,precision_score, recall_score,
from sklearn.pipeline import Pipeline
```

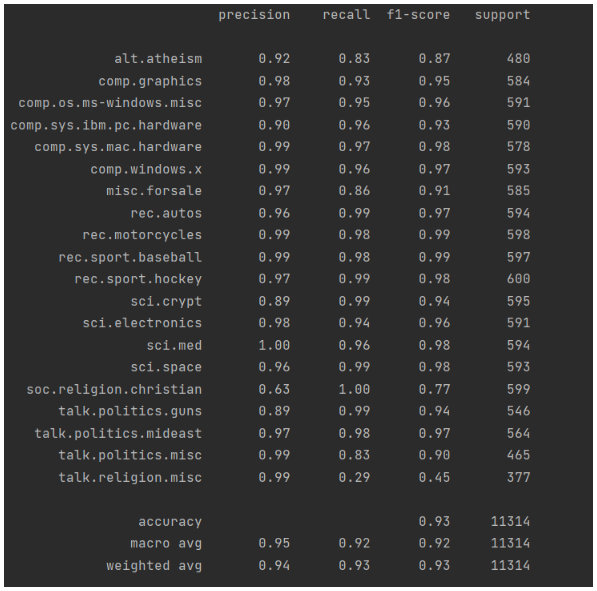
We used these packages mostly from sklearn library in order to use in related tasks. For instance metrics package was used in the f1 score, recall score and precision score's output values' visualization. For naive bayes algorithm, we imported MultinomialNB from sklearn's naive bayes library. Other classifiers' packages were imported from sklearn as well. Also, we used pipeline from sklearn's pipeline in order to create a pipeline to use features in our classifiers inputs.

B. Fetching Data

In order to fetch the data from the internet, we used ”twenty_train = fetch_20newsgroups(subset=’train’, shuffle=True, random_state=42)”. We could also use the dataset locally after downloading it directly to our computer.

C. Naive Bayes Classifier

For using the Naive Bayes Classifier, first we imported the MultinominalNB from sklearn. After that we created a pipeline with count vectorizer to convert the text into numerical format, tf-idf to weigh words based on their significance in the document and naive bayes. After that we calculated the precision score, recall score, f1 score and accuracy score for the naive bayes classifier.

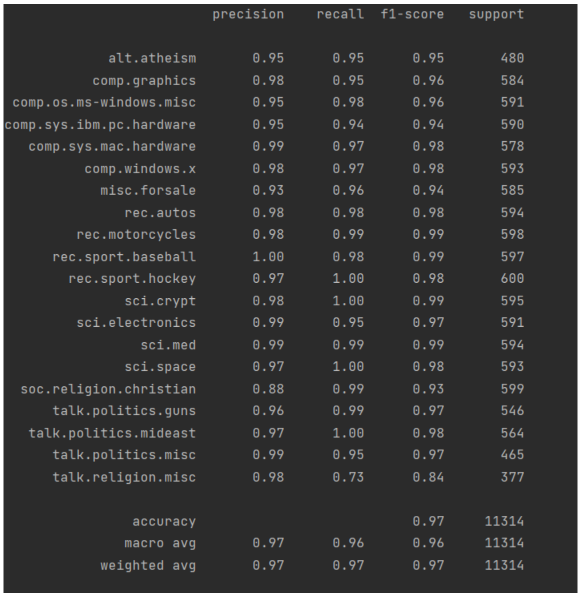


```
                          precision    recall  f1-score   support

             alt.atheism       0.92      0.83      0.87       480
           comp.graphics       0.98      0.93      0.95       584
 comp.os.ms-windows.misc       0.97      0.95      0.96       591
comp.sys.ibm.pc.hardware       0.90      0.96      0.93       590
   comp.sys.mac.hardware       0.99      0.97      0.98       578
          comp.windows.x       0.99      0.96      0.97       593
            misc.forsale       0.97      0.86      0.91       585
               rec.autos       0.96      0.99      0.97       594
         rec.motorcycles       0.99      0.98      0.99       598
      rec.sport.baseball       0.99      0.98      0.99       597
        rec.sport.hockey       0.97      0.99      0.98       600
               sci.crypt       0.89      0.99      0.94       595
         sci.electronics       0.98      0.94      0.96       591
                 sci.med       1.00      0.96      0.98       594
               sci.space       0.96      0.99      0.98       593
  soc.religion.christian       0.63      1.00      0.77       599
      talk.politics.guns       0.89      0.99      0.94       546
   talk.politics.mideast       0.97      0.98      0.97       564
      talk.politics.misc       0.99      0.83      0.90       465
      talk.religion.misc       0.99      0.29      0.45       377

                accuracy                           0.93     11314
               macro avg       0.95      0.92      0.92     11314
            weighted avg       0.94      0.93      0.93     11314
```

Fig. 1. Naive Bayes Report

D. Stochastic Gradient Descent Classifier

For using the Stochastic Gradient Descent Classifier, first we imported the SGDClassifier from sklearn. After that we created a pipeline with count vectorizer to convert the text into numerical format, tf-idf to weigh words based on their significance in the document and sgdc. After that we calculated the precision score, recall score, f1 score and accuracy score for the Stochastic Gradient Descent Classifier.

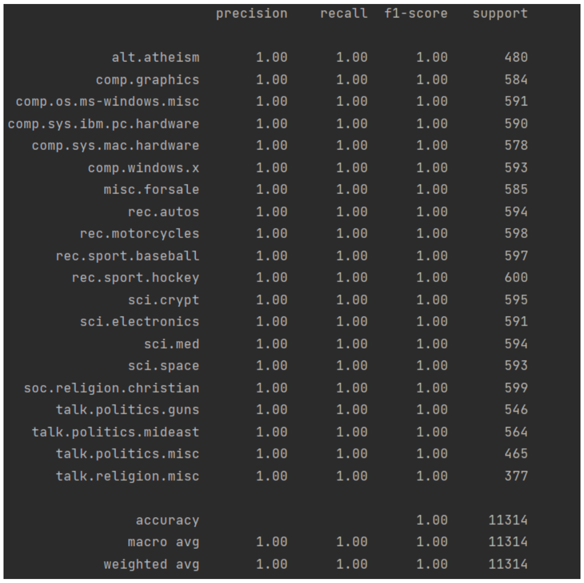


```
                          precision    recall  f1-score   support

             alt.atheism       0.95      0.95      0.95       480
           comp.graphics       0.98      0.95      0.96       584
 comp.os.ms-windows.misc       0.95      0.98      0.96       591
comp.sys.ibm.pc.hardware       0.95      0.94      0.94       590
   comp.sys.mac.hardware       0.99      0.97      0.98       578
          comp.windows.x       0.98      0.97      0.98       593
            misc.forsale       0.93      0.96      0.94       585
               rec.autos       0.98      0.98      0.98       594
         rec.motorcycles       0.98      0.99      0.99       598
      rec.sport.baseball       1.00      0.98      0.99       597
        rec.sport.hockey       0.97      1.00      0.98       600
               sci.crypt       0.98      1.00      0.99       595
         sci.electronics       0.99      0.95      0.97       591
                 sci.med       0.99      0.99      0.99       594
               sci.space       0.97      1.00      0.98       593
  soc.religion.christian       0.88      0.99      0.93       599
      talk.politics.guns       0.96      0.99      0.97       546
   talk.politics.mideast       0.97      1.00      0.98       564
      talk.politics.misc       0.99      0.95      0.97       465
      talk.religion.misc       0.98      0.73      0.84       377

                accuracy                           0.97     11314
               macro avg       0.97      0.96      0.96     11314
            weighted avg       0.97      0.97      0.97     11314
```

Fig. 2. Report of SGDC

E. Decision Tree Classifier

For using the Decision Tree Classifier, first we imported the DecisionTreeClassifier from sklearn. After that we created a pipeline with count vectorizer to convert the text into numerical format, tf-idf to weigh words based on their significance in the document and dtc. After that we calculated the precision score, recall score, f1 score and accuracy score for the Decision Tree Classifier.



```
                          precision    recall  f1-score   support

             alt.atheism       1.00      1.00      1.00       480
           comp.graphics       1.00      1.00      1.00       584
 comp.os.ms-windows.misc       1.00      1.00      1.00       591
comp.sys.ibm.pc.hardware       1.00      1.00      1.00       590
   comp.sys.mac.hardware       1.00      1.00      1.00       578
          comp.windows.x       1.00      1.00      1.00       593
            misc.forsale       1.00      1.00      1.00       585
               rec.autos       1.00      1.00      1.00       594
         rec.motorcycles       1.00      1.00      1.00       598
      rec.sport.baseball       1.00      1.00      1.00       597
        rec.sport.hockey       1.00      1.00      1.00       600
               sci.crypt       1.00      1.00      1.00       595
         sci.electronics       1.00      1.00      1.00       591
                 sci.med       1.00      1.00      1.00       594
               sci.space       1.00      1.00      1.00       593
  soc.religion.christian       1.00      1.00      1.00       599
      talk.politics.guns       1.00      1.00      1.00       546
   talk.politics.mideast       1.00      1.00      1.00       564
      talk.politics.misc       1.00      1.00      1.00       465
      talk.religion.misc       1.00      1.00      1.00       377

                accuracy                           1.00     11314
               macro avg       1.00      1.00      1.00     11314
            weighted avg       1.00      1.00      1.00     11314
```

Fig. 3. Report of DTC

We figured that the DTC was the best classifier for our dataset training task. So, we are going to move to the next step with this classification algorithm.

F. Lowercasing

As mentioned above, we decided to move forward with the DTC. For lowercase feature the default value is true. However, we tried both lowercase=True and lowercase=False values as CountVectorizer's attributes.

See Fig. 4. for the screenshot.

G. StopWords

For stop words, we tried both stop_words='english' value as CountVectorizer's attributes.

See Fig. 4. for the screenshot.

H. Analyzer in combination with ngram_range

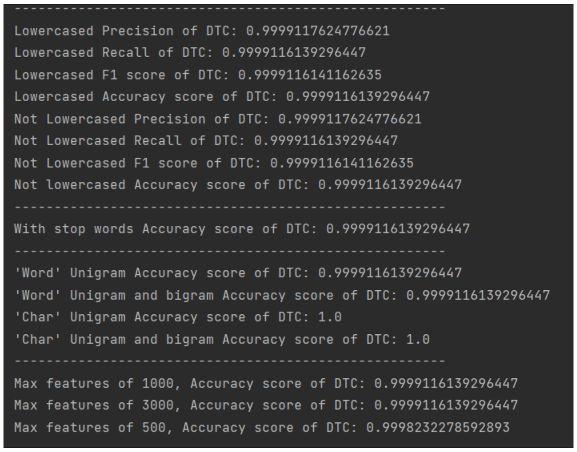
For analyzer we assigned two different analyzer values as 'word' and 'char'. Also we combined it with unigram range values with unigram to unigram (1, 1) and unigram to bigram (1, 2).

See Fig. 4. for the screenshot.

I. Maximum Features

For maximum features, we assigned max_features value of the count vectorizer to 500, 1000 and 3000. As result as max_features value increase the accuracy increases.

See Fig. 4. for the screenshot.

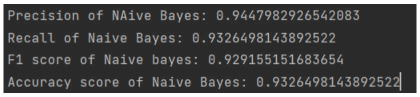


```
------------------------------------------------------
Lowercased Precision of DTC: 0.9999117624776621
Lowercased Recall of DTC: 0.9999116139296447
Lowercased F1 score of DTC: 0.9999116141162635
Lowercased Accuracy score of DTC: 0.9999116139296447
Not Lowercased Precision of DTC: 0.9999117624776621
Not Lowercased Recall of DTC: 0.9999116139296447
Not Lowercased F1 score of DTC: 0.9999116141162635
Not lowercased Accuracy score of DTC: 0.9999116139296447
------------------------------------------------------
With stop words Accuracy score of DTC: 0.9999116139296447
------------------------------------------------------
'Word' Unigram Accuracy score of DTC: 0.9999116139296447
'Word' Unigram and bigram Accuracy score of DTC: 0.9999116139296447
'Char' Unigram Accuracy score of DTC: 1.0
'Char' Unigram and bigram Accuracy score of DTC: 1.0
------------------------------------------------------
Max features of 1000, Accuracy score of DTC: 0.9999116139296447
Max features of 3000, Accuracy score of DTC: 0.9999116139296447
Max features of 500, Accuracy score of DTC: 0.9998232278592893
```

Fig. 4. Question 4

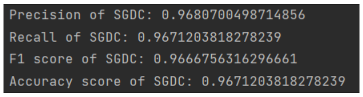
IV. Conclusion

As a result, the best results we got with the data sets we have; When we compared Naive Bayes Classifier, Stochastic Gradient Classifier and Decision Tree Classifier algorithms, we saw that we got the best result with Decision Tree Classifier.



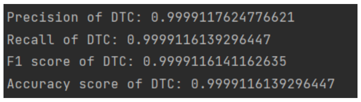
```
Precision of NAive Bayes: 0.9447982926542083
Recall of Naive Bayes: 0.9326498143892522
F1 score of Naive bayes: 0.929155151683654
Accuracy score of Naive Bayes: 0.9326498143892522
```

Fig. 5. Metrics output of Naive Bayes Classifier



```
Precision of SGDC: 0.9680700498714856
Recall of SGDC: 0.9671203818278239
F1 score of SGDC: 0.9666756316296661
Accuracy score of SGDC: 0.9671203818278239
```

Fig. 6. Metrics Output of Stochastic Gradient Descent Classifier



```
Precision of DTC: 0.9999117624776621
Recall of DTC: 0.9999116139296447
F1 score of DTC: 0.9999116141162635
Accuracy score of DTC: 0.9999116139296447
```

Fig. 7. Metrics Output of Decision Tree Classifier

References

[1] YouTube https://www.youtube.com/watch?v=0kPRaYSgblM&ab_channel=CodeWrestling.

[2] Scikit Documentation. https://scikit-learn.org/stable/tutorial/text_analytics/working_with_text_data.html#loading-the-20-newsgroups-dataset.

[3] Scikit Documentation. https://scikit-learn.org/stable/modules/generated/sklearn.feature_extraction.text.CountVectorizer.html.

[4] Scikit Documentation for DTC. https://scikit-learn.org/stable/modules/generated/sklearn.tree.DecisionTreeClassifier.html.

[5] Medium Article. https://chirag-sehra.medium.com/decision-trees-explained-easily-28f23241248#:~:text=A%20decision%20tree%20classifier%20is,belonngs%20to%20the%20same%20class..

[6] Medium Article. https://shorturl.at/bmoDR.

[7] Text Classification Methods. https://monkeylearn.com/text-classification/.