MIDAS@IIITD MULTIMODAL DIGITAL MEDIA ANALYSIS LAB

SUMMER INTERNSHIP/RA TASK 2021

Task opted – *Task 3 NLP*

README

Ajay Agarwal
Dept. of Computer Science & Engineering
DIT University

Approach –

I began by considering all the questions asked in the document and conducting a simple introspective survey of what toolkits might be required and what process I must require to solve the cases. Hence, I divided my entire work journey into 5 simple subtasks. Each subtask had its steps that must be completed.

SUBTASKS :

1. Cleaning the data
2. Data Visualization and Pre-Processing
3. Determining Accuracy Metrics
4. Trying various ML Classification Algorithms
5. Identification of strategies that might improve classification accuracy further

Let’s begin by breaking down the steps required for each sub-task

SUBTASK 1. CLEANING THE DATA

1. Understanding the data and performing basic EDA (Exploratory Data Analysis)
2. Understanding the Product Category Tree structure
3. Finding out the maximum and minimum number of sub-categories in the category tree
4. Separating all the sub-categories
5. Identification of “primary category”

SUBTASK 2(a) DATA VISUALIZATION

1. Removing all the columns except "description" and "first category"
2. Visualization of "first category" columns to identify category distribution
3. Assessing possibilities to manually merging "wordy" first category (that earlier had no further sub-categories) into existing parent category
4. If step 3 prevails, visualization of merged category distribution
5. If step 3 fails, dropping all such categories.
6. Wordcloud visualization for "description" column
7. Providing a statistical EDA of training dataset
8. Visualization of description length variation

SUBTASK 2(b) PRE-PROCESSING AND PREPARING TRAINING DATASET

1. Identification of (if any) null values
2. Training & Testing dataset split (70:30)
3. Stopword removal from Description Column
4. Tf-IDF vectorization with Pipeline formation (along with LinearSVC)

Question – Why did I choose Term Frequency Inverse Document Frequency over CountVectorizer?

Answer – Partially, because Tf-IDF is usually faster on CPUs and provides better results. I am working on a CPU.

SUBTASK 3 IDENTIFICATION OF ACCURACY METRICS

The following accuracy metrics were measured –

1. Weighted Precision
2. Weighted Accuracy
3. Weighted Recall
4. F1 score
5. Corresponding Confusion Matrix was plotted

SUBTASK 4. TRYING OUT VARIOUS ML CLASSIFICATION ALGORITHMS

1. Linear Support Vector Machine was used
2. Naïve Bayes was used
3. KNN Classifier was used
4. Random Forest Classifier was used.
5. Check for over-fitting (if any) using K-fold cross-validation score

SUBTASK 5. IDENTIFICATION OF STRATEGIES THAT MIGHT IMPROVE CLASSIFICATION ACCURACY

1. Pre-processing with Word2Vec and utilizing Logistic Regression
2. Pre-processing with Doc2Vec and utilizing Logistic Regression
3. Pre-processing with Bag-of-Words and utilizing Keras

Question – Can accuracy be *actually* improved?

Answer – *Most likely, no.*

There are two reasons for this answer –

1. Last year, I tried a Kaggle competition on identifying the news category based on headlines and short descriptions [1]. Personal insights from the competition and studying solutions provided by other Kaggle experts provided similar conclusions i.e. Word Embedding or heavy-architecture enabled BERT or distilBERT models might appear attractive for this use-case, however, in reality, they perform poorly against simple ML classifiers like LR or Linear SVMs
2. Still, I tried (*not as part of the submission)* the second suggestion I posed – preprocessing with Doc2Vec with LR and I achieved an accuracy of 66.38% which was much worst than the most inferior performing ML algorithm that I tried. For implementing Doc2Vec and other details, some references were used. [2-3].

OTHER DETAILS

1. The experimental Log file has been added separately as a .txt that documents all the errors encountered in the due process along with the solutions found for the same.
2. README.pdf is being added.
3. The file requiements.txt has been added.

BENCHMARK RESULTS FOR THE GIVEN TASK

The accuracy metrics were Accuracy (Acc), Weighted Precision (P), Weighted Recall (R), and F1 score (F1) for the models – Linear SVM, KNN, Naïve Bayes, and Random Forest

Table 1. depicts the performance of ML algorithms for the given dataset.

Model	**Accuracy**	**P**	**R**	**F1**
LinearSVC	**97.78**	**97.70**	**97.78**	**97.69**
KNN	95.16	95.09	95.16	95.06
Naïve Bayes	82.19	83.19	82.19	78.99
Random Forest	94.97	94.82	94.97	94.65

Table 1. Accuracy, Weighted Precision, Weighted Recall, and F1 score of ML classifiers on the given dataset.

Hence, our benchmark ML classifier is Linear Support Vector Machine with an F1 score of 97.69% followed closely by KNN with an F1 score of 95.06%

REFERENCES

1. https://www.kaggle.com/rmisra/news-category-dataset/
2. https://towardsdatascience.com/text-classification-with-nlp-tf-idf-vs-word2vec-vs-bert-41ff868d1794
3. https://www.kdnuggets.com/2018/11/multi-class-text-classification-doc2vec-logistic-regression.html#:~:text=DBOW%20is%20the%20doc2vec%20model,sampled%20word%20from%20the%20paragraph.