IOANNA PAPAGIANNI 2790 – CHRISTOS GOULAS 2677

QUESTION 5 (BONUS)

For this analysis we use Jupyter Notebook and the file is in IPython format. We downloaded "train.csv" and "test.csv" from Kaggle and all files are in the same folder with "fakenews_disaster_detection_1.ipynb" (we submitted two more different editions of our code "fakenews_disaster_detection_2.ipynb" and "fakenews_disaster_detection_3.ipynb" but the score was lower than the first.

"fakenews_disaster_detection_1.ipynb" :
Our public score with Multinomial Naive-Bayes, TfidfVectorizer, 'text' column used: 0.80265
Our public score with Multinomial Naive-Bayes, CountVectorizer, 'text' column used: 0.79550
"fakenews_disaster_detection_2.ipynb":
Our public score with Multinomial Naive-Bayes TfidfVectorizer, 'text', 'keyword', 'location' : 0.52249
Our public score with Multinomial Naive-Bayes TfidfVectorizer, 'text', 'location' : 0.52453
"fakenews_disaster_detection_3.ipynb":
Our public score with linearSVM, TfidfVectorizer, 'text' column used: 0.78425

In our first approach, we experiment with multinomial Naive Bayes and two different vectorizers. The multinomial Naive Bayes classifier is suitable for classification with discrete features and it performs well with text classification. The multinomial distribution normally requires integer feature counts. However, in practice, fractional counts such as TfidfVectorizer work but do not perform as well as CountVectorizer and in this problem TfidfVectorizer performs better because in every Tweet we have a limit of characters.Note that, in "fakenews_disaster_detection_1.ipynb" we use for both training and testing only one column ('text').

In our second approach, the difference is that now we also use 'keyword' and 'location' column for both training and testing. Surprisingly, we score lower than in our first approach. The reason might be that 'keyword' is repeated also in the text in the most tweets and it gets biased with TfidfVectorizer.
To avoid the bias we mentioned before, then we take into account only 'location' and 'text' column. Our score is better than our second approach but still lower than our first.

In our third approach, we try Linear Support Vector Classification (linear SVC) that has more flexibility in the choice of penalties and loss functions and should scale better to large numbers of samples but does not perform better than MultinomialNB in our case. It represents data as points in space and tries to create a mapping with a wide as possible gap between the separate categories. Thus SVM can efficiently handle sparse data but again, not in the level of MultinomialNB. Note that, in "fakenews_disaster_detection_3.ipynb" we use for both training and testing only one column ('text').

KAGGLE NAMES: GoulasAC, JohannaP
KAGGLE TEAM NAME: Goulas-Papagianni
KAGGLE SCORE: 0.80265