Methodological approach:

1. We exported the data from the CSV files and cleaned it (dropping NAs).
2. Stored the data as tokens and labels in dataframe
3. We used the label encoder to encode the y labels
4. Created the x_train, y_train, x_test, y_test
5. We tried between CountVectorizer and TfIDFVectorizer and the latter gave better results hence we moved ahead with it
    a. CountVectorizer – We used it to find the frequency of each word that occurs in the entire text
    b. TfIDFVectorizer – On top of finding the word's frequency, we felt it was important to find the relative frequency of any given term concerning the corpus.
6. We tried using multiple feature extraction methods such as:
    a. N-Grams – We used N-grams to mine the patterns of the co-occurring words which might be useful for classification of text into BIO
    b. Token Statistics (with and without POS) – We used the the token frequency and token length
    c. POS (with and without POS) – We used POS because they are helpful in building parse trees and extracting relations between words

    We got the best results with N-grams (F1-Score – Macro: 0.4426 & F1-Score – Weighted: 0.6693)

7. Classifier: We tried using multiple classifiers such as
    a. ADABoostClassifier
    b. Naïvebayes Classifier
    c. SVM

    As mentioned in the lecture slides we tried different ML models and out of the three, ADABoostClassifier performed the best, with the highest F1 scores. These scores were attained by using just N-grams. We tried using Token Statistics with POS to get better accuracy but only resulted in lower scores.