Problem Statement

**Custom message/phrase detection for reminders Task**:

Given a query related reminders, detect the custom message/phrase that user wants to set the reminder for (if present).

Eg. Query: Please remind me to go to gym.

Reminder phrase: “go to gym”

Solution

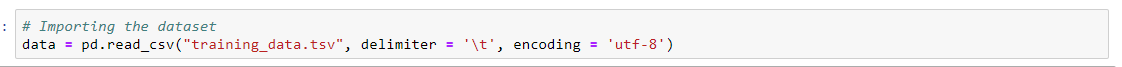
We will follow 2 approaches in building this model. Link of jupyter notebook file for both approaches is included with this assignment.

Approach 1: Using TextBlob

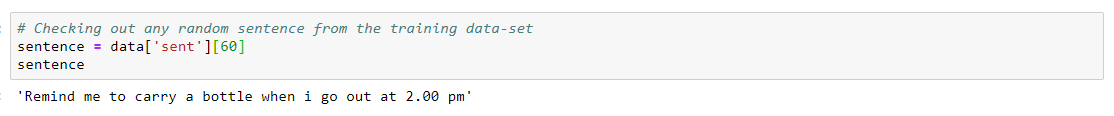
Importing all libraries and packages required



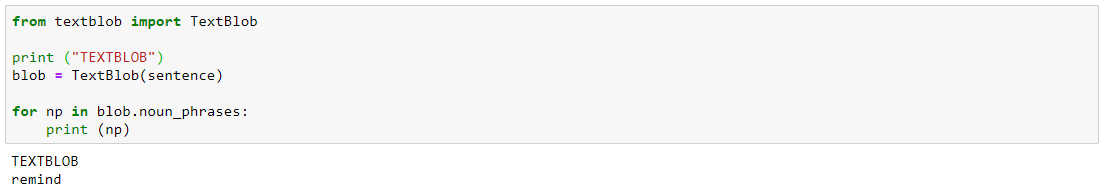
Importing the dataset



Checking out any random sentence from the training data-set



Applying the TextBlob function

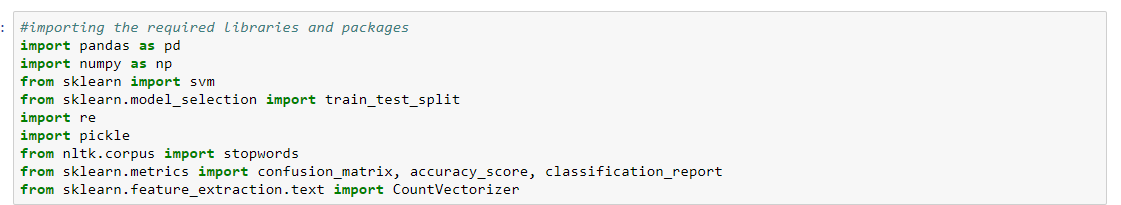


As we can see that the results are very unsatisfactory, so we cannot use this approach. We will now use Approach 2 in which we will build a Support Vector Machine Model.

Approach 2: Support Vector Machine for NLP

Part 1: Training and testing the dataset

Importing the required libraries and packages



Importing the dataset



All the values in the column ‘label’ will go in dataframe x and all the values in the colum n ‘sent’ will go in dataframe y

Processing data in x and y



We converted x and y in series for splitting

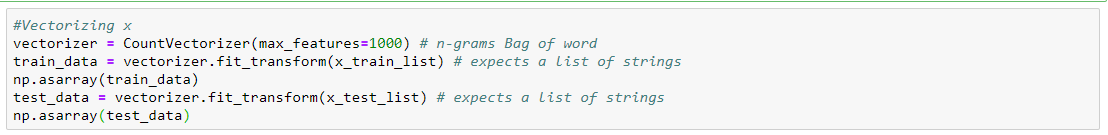
Splitting the data into train and test



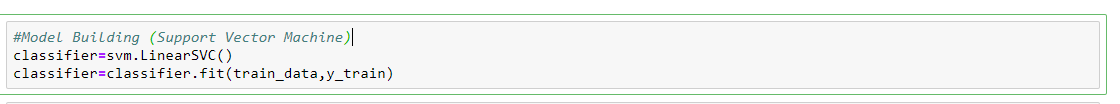
Making list of x\_train and x\_test in order to vectorise it in the next step



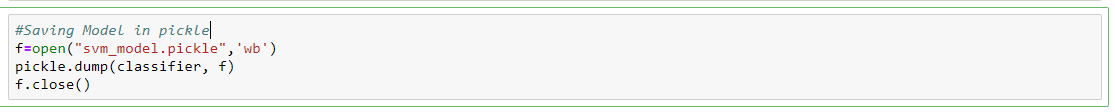
Vectorizing x\_train and x\_test



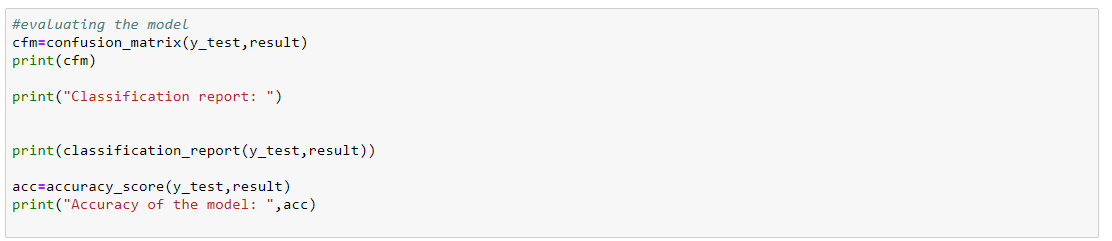
Now we will build our SVM Model



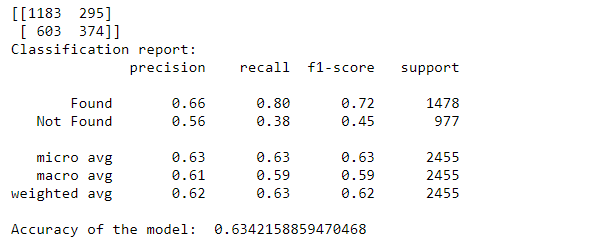
We will save this model as ‘svm\_model\_pickle’



Evaluation of the model



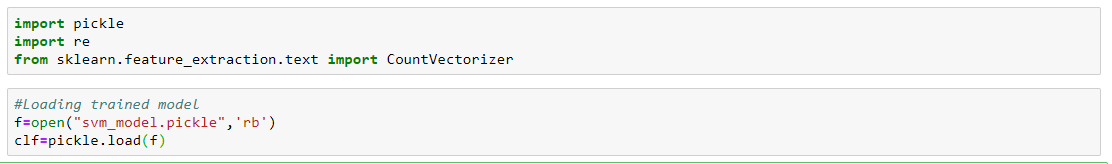
Output



Conclusion: We have built a decent SVM Model with an accuracy of 63%. However there are a lot of type 1 and type 2 errors.

Part 2: Testing the 'svm\_model' on 'eval\_data.text'

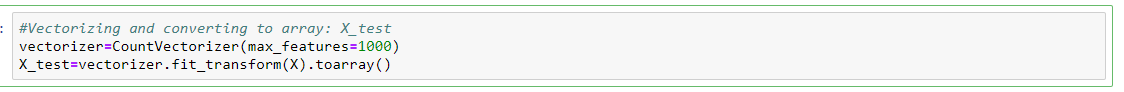
Importing the libraries and our saved model ‘svm\_model’



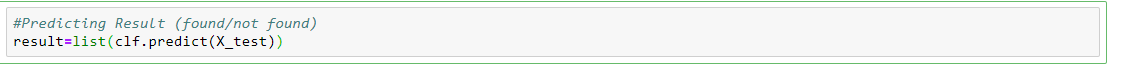
Importing the ‘eval\_data\_text’ file on which we have to perform our predictions



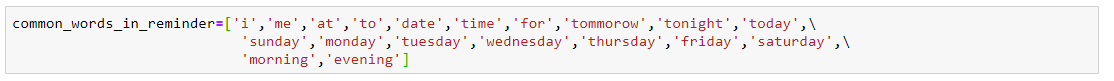
Now we will vectorize and convert x\_test into array for fitting it to the model



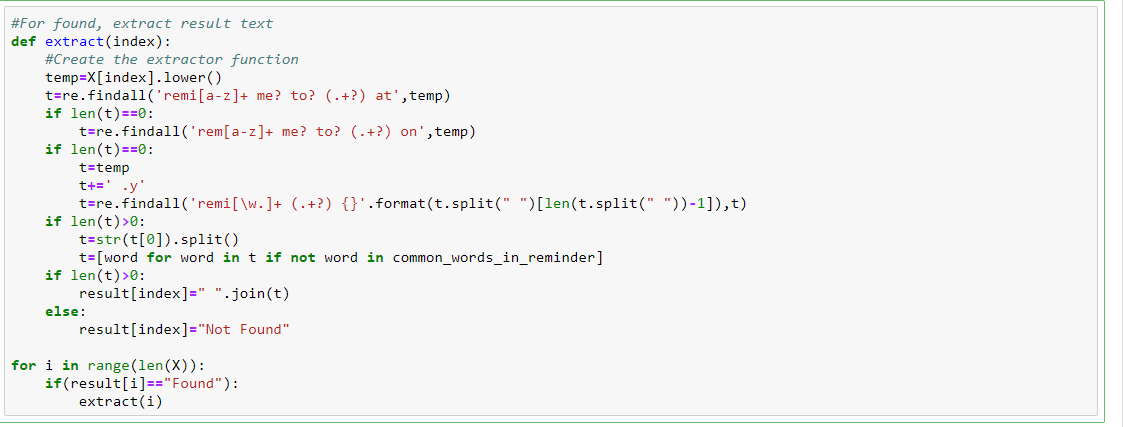
Now we will predict whether the word was found or not found in the data



No we will create a list of common words that are most commonly present in a reminder



No if the above words are found we will define a function which will extract those words and append to the data



Now we will save the final result as a tsv file



So now are model is complete, let’s look at some results from the output file











Thus looking at the results we can conclude that our model is not the best model. But it is a decent model and can be considered as a good base to start of with in the future.