ENGR 102 – Programming Practice

Practice Session (Week 14)

In this practice session, you will build a program with a graphical user interface that will try to predict the emotional meaning of a sentence or a word that the user enters. Using your classification skills that you learned in class, you will build a classifier and train that classifier with given dataset then you will try to predict the emotions of user provided words.

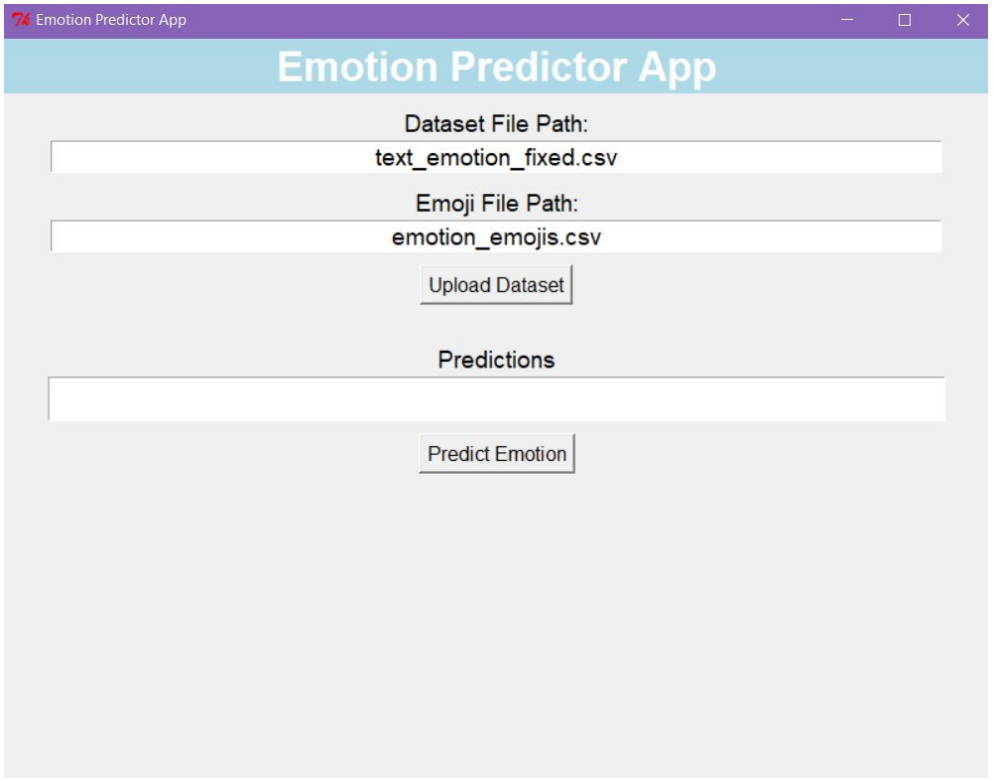

Figure 1 (initial GUI)

Datasets:

As part of the practice session material, you are provided with two different datasets. One ("text_emotion_fixed.csv") is for training the classifier, which is formatted as (emotion, document), and the other ("emotion_emojis.csv") is for getting proper emojis for emotions, which is formatted as (emotion,emoji).

How does it work?

Firstly, implement the interface in Figure 1. When the user press "Upload Dataset" button, you will parse the data appropriately and create a classifier. Then, train that classifier with the data coming from first dataset, and create a dictionary for mapping emotion to emojis for further use. After these steps finish, the user will enter a word or sentence through an entry widget labeled as "Predictions", and press "Predict Emotion" button. When the user presses "Predict Emotion" button, you will try to predict the most likely and least likely 3 emotions with the methods in docclass.py, and show those emotions to the user with proper emojis which comes from the second dataset (see Figure 2).

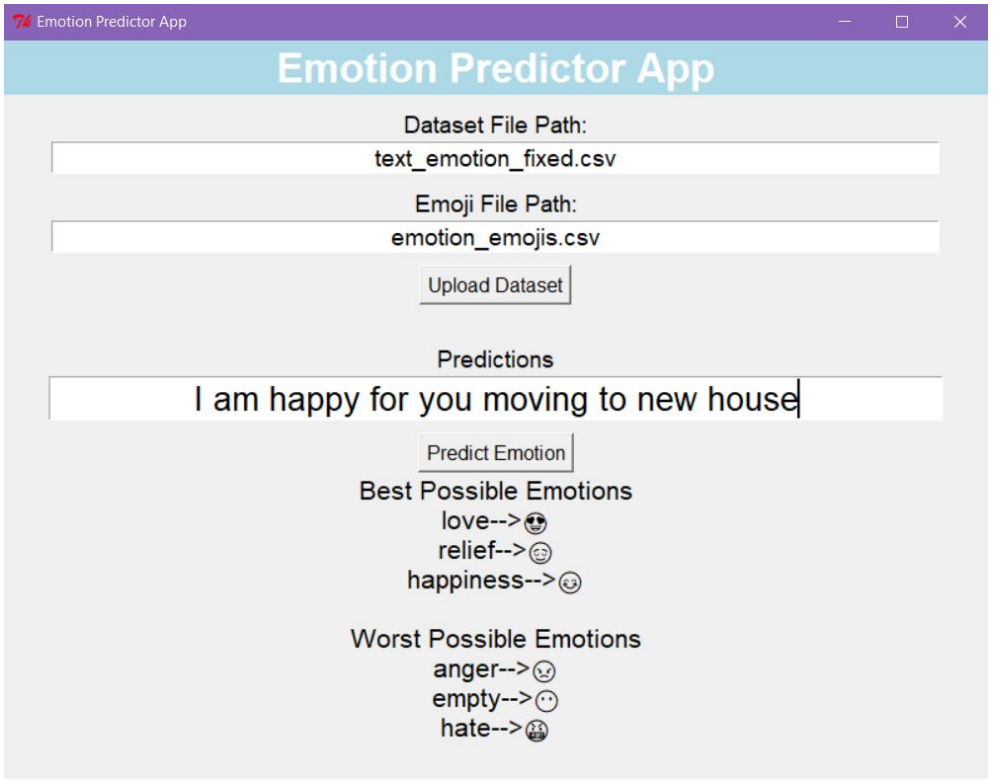

Figure 2

Implementation Notes:

- For the classifier, you need to import docclass.
- For the emojis, you need to install "emoji" library and import it.
- For adding emojis to label; emojis are unicode too, so you can behave them as string. In the dataset you have CLDR codes of emojis. For creating unicoded emojis from CLDR code, you need to use emojize method from emoji library. For example;

```
import emoji
emoji_cldr = "smiling_face_with_smiling_eyes"
emoji_unicode = emoji.emojize(":%s:"%emoji_cldr)
print(emoji_unicode, type(emoji_unicode))

>>> 😊 <type 'unicode'>
```