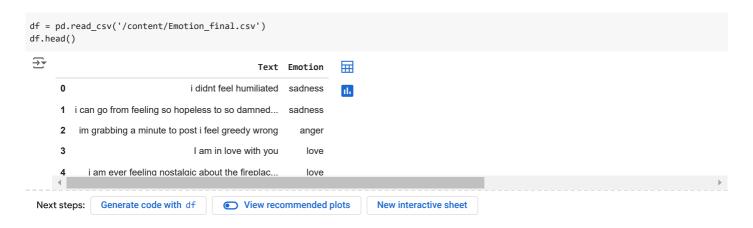
```
from google.colab import drive
drive.mount('/content/drive')

Drive already mounted at /content/drive; to attempt to forcibly remount, call drive.mount("/content/drive", force_remount=True).

importing necessary libraries

import pandas as pd
import numpy as np
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

#### step 1 : load the dataset into a dataframe



# step 2 : Perform the data cleaning



```
# checking for missing values

df.isnull().sum()

Text 0
Emotion 0
dtype: int64

df.isnull().sum()/len(df)

Text 0.0
Emotion 0.0
dtype: float64
```

#### b.remove duplicates

```
# drop rows with missing values
df.dropna(subset=['Text', 'Emotion'], inplace = True)

# check duplicate rows
df.duplicated().sum()

# remove duplicate rows
df.drop_duplicates(inplace=True)

# Reset index after dropping rows
df.reset_index(drop=True, inplace=True)
```

# step 3 : Label encoding the emotion column

#### step 4: train a random forest model with the dataset

```
# define x and y variables
X = df['Text']
Y = df['Emotion']
#split a data set into train and test
from sklearn.model_selection import train_test_split
X_train,X_test,Y_train,Y_test = train_test_split(X, Y, train_size = 0.2, random_state = 42)
from sklearn.ensemble import RandomForestClassifier
from sklearn.feature_extraction.text import TfidfVectorizer
# Create a TfidfVectorizer to convert text to numerical features
vectorizer = TfidfVectorizer()
# Fit the vectorizer on the training data and transform both training and testing data
X_train_vec = vectorizer.fit_transform(X_train)
X_test_vec = vectorizer.transform(X_test)
# Now fit the model with the vectorized data
model = RandomForestClassifier(n_estimators=100, random_state=42)
model.fit(X_train_vec, Y_train)
               RandomForestClassifier
     RandomForestClassifier(random_state=42)
```

## Step 5: Find the accuracy of the model

```
# predictions
Y_pred = model.predict(vectorizer.transform(X_test)) # Transform x_test before prediction
from sklearn.metrics import accuracy_score
# evaluating the model
accuaracy = accuracy_score(Y_test, Y_pred)
print("accuracy:" , accuaracy)
→ accuracy: 0.7259771986970684
from sklearn.metrics import confusion_matrix, classification_report
# Calculate confusion matrix (use a different variable name)
conf_matrix_result = confusion_matrix(Y_test, Y_pred)
print("confusion_matrix:" , conf_matrix_result)
                            9 0 75 440
0 3 0]
→ confusion_matrix: [[1579
                                               4 274
        0
              0 2 6
              0 1298 483 10 258 28]
0 43 5060 58 421 14]
        54
        43
        19
              0
                   7 622 570 111
                                     2]
              0 81 1133 13 3675
        89
              0 108 186
#classification_report
c_report = classification_report(Y_test, Y_pred)
print("classification_report:" , c_report)
🚁 /usr/local/lib/python3.10/dist-packages/sklearn/metrics/_classification.py:1471: UndefinedMetricWarning: Precision and F-score are
       _warn_prf(average, modifier, msg_start, len(result))
     classification_report:
                                         precision
                                                      recall f1-score support
                        0.88
                                 0.66
                                           0.76
                                                     2375
            anger
        confusion
                                 0.00
                        0.00
                                                       11
            fear
                       0.80
                                 0.61
                                           0.69
                                                     2131
                                 0.90
                                           0.75
                                                     5639
            happy
                       0.64
                                 0.43
                                           0.57
                       0.87
                                                     1331
            love
                                 0.74
         sadness
                       0.76
                                           0.75
                                                     4999
         surprise
                       0.84
                                 0.42
                                           0.56
                                                     706
         accuracy
                                           0.73
                                                    17192
                        0.68
                                 0.54
                                           0.58
                                                    17192
        macro avg
     weighted avg
                       0.75
                                 0.73
                                           0.72
                                                    17192
     /usr/local/lib/python3.10/dist-packages/sklearn/metrics/_classification.py:1471: UndefinedMetricWarning: Precision and F-score are
       _warn_prf(average, modifier, msg_start, len(result))
     /usr/local/lib/python3.10/dist-packages/sklearn/metrics/_classification.py:1471: UndefinedMetricWarning: Precision and F-score are
       _warn_prf(average, modifier, msg_start, len(result))
```