In [1]:

```
import pandas as pd
import pickle
```

In [2]:

```
df=pd.read_csv("C:/Users/akil/Desktop/Model Training process-2/new_data_with_sen
df=df[['text','new_label']]
df.dropna(inplace=True)
df.drop_duplicates(inplace=True)
```

In [3]:

```
label_mapping = {'LABEL_0': 0, 'LABEL_1': 1, 'LABEL_2': 2}  # Modify based on yo
df['new_label'] = df['new_label'].map(label_mapping)
```

In [4]:

```
df
```

Out[4]:

	text	new_label
0	@switchfoot http://twitpic.com/2y1zl - Awww, t...	0
1	is upset that he can't update his Facebook by ...	0
2	@Kenichan I dived many times for the ball. Man...	1
3	my whole body feels itchy and like its on fire	0
4	@nationwideclass no, it's not behaving at all....	0
...	...	...
2000053	We bought this Thomas for our son who is a hug...	1
2000054	My son recieved this as a birthday gift 2 mont...	0
2000055	I bought this toy for my son who loves the "Th...	0
2000056	This is a compilation of a wide range of Mitfo...	2
2000057	This DVD will be a disappointment if you get i...	0

1981441 rows × 2 columns

In [6]:

```
import re
import nltk
from nltk.tokenize import word_tokenize
from nltk.corpus import stopwords
from nltk.stem import WordNetLemmatizer
nltk.download("stopwords")
nltk.download("punkt")
nltk.download("wordnet")
```

```
[nltk_data] Downloading package stopwords to
[nltk_data]     C:\Users\akil\AppData\Roaming\nltk_data...
[nltk_data]   Package stopwords is already up-to-date!
[nltk_data] Downloading package punkt to
[nltk_data]     C:\Users\akil\AppData\Roaming\nltk_data...
[nltk_data]   Package punkt is already up-to-date!
[nltk_data] Downloading package wordnet to
[nltk_data]     C:\Users\akil\AppData\Roaming\nltk_data...
[nltk_data]   Package wordnet is already up-to-date!
```

Out[6]:

```
True
```

In [7]:

```
class Preprocessor:
    def __init__(self):
        self.stop_words = set(stopwords.words("english"))
        self.lemmatizer = WordNetLemmatizer()
        self.regex_pattern = re.compile(r"http\S+|www\S+|@\w+|#\w+|[^\w\s]|\d+")

    def clean_text(self, text):
        text = text.lower()
        text = self.regex_pattern.sub("", text)
        tokens = word_tokenize(text)
        cleaned_tokens = []
        negate = False
        negation_words = {"not", "no", "never", "n't"}
        for word in tokens:
            if word in negation_words:
                negate = True
            elif negate:
                cleaned_tokens.append(
                    "not_" + self.lemmatizer.lemmatize(word)
                )
                negate = False
            elif word not in self.stop_words:
                cleaned_tokens.append(self.lemmatizer.lemmatize(word))

        return " ".join(cleaned_tokens)
```

In [8]:

```
preprocessor = Preprocessor()
text = "I do not like movie at all! It was horrible 😡"
print(preprocessor.clean_text(text))
```

```
not_like movie horrible
```

In [9]:

```
df['cleaned_text']=df['text'].apply(preprocessor.clean_text)
```

In [11]:

```
df
```

Out[11]:

	text	new_label	cleaned_text
0	@switchfoot http://twitpic.com/2y1zl - Awww, t...	0	thats bummer shoulda got david carr third day
1	is upset that he can't update his Facebook by ...	0	upset cant update facebook texting might cry r...
2	@Kenichan I dived many times for the ball. Man...	1	dived many time ball managed save rest go bound
3	my whole body feels itchy and like its on fire	0	whole body feel itchy like fire
4	@nationwideclass no, it's not behaving at all....	0	not_it not_behaving im mad cant see
...	...	...	...
2000053	We bought this Thomas for our son who is a hug...	1	bought thomas son huge thomas fan huge set roo...
2000054	My son recieved this as a birthday gift 2 mont...	0	son recieved birthday gift month ago loved eve...
2000055	I bought this toy for my son who loves the "Th...	0	bought toy son love thomas toy need one batter...
2000056	This is a compilation of a wide range of Mitfo...	2	compilation wide range mitford article best sk...
2000057	This DVD will be a disappointment if you get i...	0	dvd disappointment get hoping see substantial ...

1981441 rows × 3 columns

In [10]:
```
df.to_csv("C:/Users/akil/Desktop/Model Training process-2/clean_data.csv",index=
```

In [11]:
```
df=pd.read_csv("C:/Users/akil/Desktop/Model Training process-2/clean_data.csv")#
df = df.dropna(subset=['cleaned_text'])  # Remove rows where 'cleaned_text' is N
```

In [12]:
```
from sklearn.model_selection import train_test_split
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score,classification_report
```

In [13]:
```
X_train, X_test, y_train, y_test = train_test_split(df['cleaned_text'], df['new_

vectorizer = TfidfVectorizer(max_features=200000)  # Limit features to 200K word
X_train_tfidf = vectorizer.fit_transform(X_train)
X_test_tfidf = vectorizer.transform(X_test)
```

In [ ]:
```
# Train a Logistic Regression model
model = LogisticRegression(max_iter=500, class_weight='balanced')  # Balanced ha
model.fit(X_train_tfidf, y_train)

# Make predictions
y_pred = model.predict(X_test_tfidf)
```

```
# Evaluate the model
accuracy = accuracy_score(y_test, y_pred)
print(f"Accuracy: {accuracy:.4f}")


# Classification Report
print("Classification Report:")
print(classification_report(y_test, y_pred))
```

```
Accuracy: 0.7720
Classification Report:
              precision    recall  f1-score   support

           0       0.82      0.77      0.79    131875
           1       0.61      0.75      0.68    103524
           2       0.88      0.78      0.83    159462

    accuracy                           0.77    394861
   macro avg       0.77      0.77      0.77    394861
weighted avg       0.79      0.77      0.78    394861
```

In [15]:

```
with open("C:/Users/akil/Desktop/Model Training process-2/sentiment_model.pkl",
    pickle.dump(model, model_file)

with open("C:/Users/akil/Desktop/Model Training process-2/tfidf_vectorizer.pkl",
    pickle.dump(vectorizer, vectorizer_file)

print("Logistic model and vectorizer saved successfully!")
```

```
Logistic model and vectorizer saved successfully!
```

In [ ]: