

ASSIGNMENT NO:3

Name :Nikam Sakshi Jalindar

Roll No:2317088

Batch:A

```
In [1]: import pandas as pd  
import numpy as np
```

```
In [3]: # Load Swiggy Dataset  
df1 = pd.read_csv("swiggy.csv")  
  
df1 = df1[['review_description', 'rating']].copy()  
df1.dropna(inplace=True)
```

```
In [5]: # Load Zomato Dataset  
df2 = pd.read_csv("zomato.csv")  
  
df2 = df2[['review_description', 'rating']].copy()  
df2.dropna(inplace=True)
```

```
In [6]: # Sentiment Label Function  
def label_sentiment(rating):  
    if rating >= 4:  
        return "Positive"  
    elif rating <= 2:  
        return "Negative"  
    else:  
        return "Neutral"  
  
df1['sentiment'] = df1['rating'].apply(label_sentiment)  
df2['sentiment'] = df2['rating'].apply(label_sentiment)
```

```
In [9]: # Import ML Libraries  
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
```

```
In [11]: # Swiggy Model  
X_train1, X_test1, y_train1, y_test1 = train_test_split(  
    df1["review_description"],  
    df1["sentiment"],  
    test_size=0.2,  
    random_state=42  
)  
  
tfidf1 = TfidfVectorizer(stop_words="english", max_features=5000)  
X_train1_tfidf = tfidf1.fit_transform(X_train1)  
X_test1_tfidf = tfidf1.transform(X_test1)  
  
model1 = LogisticRegression(max_iter=1000)  
model1.fit(X_train1_tfidf, y_train1)  
  
y_pred1 = model1.predict(X_test1_tfidf)  
  
swiggy_accuracy = accuracy_score(y_test1, y_pred1)
```

```
print("Swiggy Accuracy:", swiggy_accuracy)
```

Swiggy Accuracy: 0.8972832988869245

In [12]:

```
# Zomato Model
X_train2, X_test2, y_train2, y_test2 = train_test_split(
    df2["review_description"],
    df2["sentiment"],
    test_size=0.2,
    random_state=42
)

tfidf2 = TfidfVectorizer(stop_words="english", max_features=5000)
X_train2_tfidf = tfidf2.fit_transform(X_train2)
X_test2_tfidf = tfidf2.transform(X_test2)

model2 = LogisticRegression(max_iter=1000)
model2.fit(X_train2_tfidf, y_train2)

y_pred2 = model2.predict(X_test2_tfidf)

zomato_accuracy = accuracy_score(y_test2, y_pred2)

print("Zomato Accuracy:", zomato_accuracy)
```

Zomato Accuracy: 0.8980632008154944

In [17]:

```
import pandas as pd
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

# Sentiment labeling function
def label_sentiment(rating):
    if rating >= 4:
        return "Positive"
    elif rating <= 2:
        return "Negative"
    else:
        return "Neutral"

# Function to calculate accuracy
def sentiment_accuracy(file_name):

    df = pd.read_csv(file_name)
    df = df[['review_description', 'rating']].copy()
    df.dropna(inplace=True)

    df['sentiment'] = df['rating'].apply(label_sentiment)

    X_train, X_test, y_train, y_test = train_test_split(
        df['review_description'],
        df['sentiment'],
        test_size=0.2,
        random_state=42
    )

    tfidf = TfidfVectorizer(stop_words="english", max_features=5000)
    X_train_tfidf = tfidf.fit_transform(X_train)
    X_test_tfidf = tfidf.transform(X_test)

    model = LogisticRegression(max_iter=1000)
    model.fit(X_train_tfidf, y_train)
```

```
y_pred = model.predict(X_test_tfidf)

accuracy = accuracy_score(y_test, y_pred)

return accuracy

# Calculate Accuracy
swiggy_acc = sentiment_accuracy("swiggy.csv")
zomato_acc = sentiment_accuracy("zomato.csv")

print("Swiggy Sentiment Accuracy : ", swiggy_acc)
print("Zomato Sentiment Accuracy : ", zomato_acc)
```

```
Swiggy Sentiment Accuracy : 0.8972832988869245
Zomato Sentiment Accuracy : 0.8980632008154944
```

In []:

In []: