In [1]:

```python
import pandas as pd
import re
```

In [3]:

```python
df = pd.read_csv("/home/admin1/Downloads/swiggy.csv")
df
```

Out[3]:

	App	review_date	review_description	rating	thumbsUpCount	developer_re
0	Swiggy	2023-07-24 09:57:40	I have been using swiggy for a long time and I...	2	103	Hey th apologize inconven
1	Swiggy	2023-07-23 10:35:23	Worst experiences I'm having with the app for ...	1	12	Hello, we wou know more
2	Swiggy	2023-07-24 14:48:26	The best foolishing app with offers. The app o...	1	7	We are sor you down wi
3	Swiggy	2023-07-07 08:26:00	Title: Disappointing Experience with Swiggy In...	1	85	Hey there sorry to have
4	Swiggy	2023-07-19 07:40:05	Worst ever experience. I ordered from instamar...	1	29	We're ap about this exp y
...	...	...	...	...	...	
200786	Swiggy	2019-01-09 08:24:54	Very Good Experience with Swiffy	5	0	Hey Navjyo pleased enjoy
200787	Swiggy	2019-11-30 15:06:54	Good offers. Best food	5	0	Hey, thank rating! This m
200788	Swiggy	2019-12-10 11:49:51	It's real awesome fast delivery	5	0	Hey, thank rating! This m
200789	Swiggy	2019-11-11 11:32:01	Super app, and healful	5	0	Hey, thank rating! This m
200790	Swiggy	2019-11-18 20:04:15	very user friendly	5	0	Hi Arnav, th for the perfe

200791 rows × 8 columns

In [5]:

```python
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 200791 entries, 0 to 200790
Data columns (total 8 columns):
 #   Column                   Non-Null Count   Dtype 
---  ------                   --------------   ----- 
 0   App                      200791 non-null  object
 1   review_date              200791 non-null  object
 2   review_description       200791 non-null  object
 3   rating                   200791 non-null  int64 
 4   thumbsUpCount            200791 non-null  int64 
 5   developer_response       197247 non-null  object
 6   developer_response_date  197247 non-null  object
 7   appVersion               170082 non-null  object
dtypes: int64(2), object(6)
memory usage: 12.3+ MB
```

In [7]:
```
df["rating"].value_counts()
```

Out[7]:
```
rating
1    101539
5     65711
4     15399
2      9410
3      8732
Name: count, dtype: int64
```

In [9]:
```
df = df[["review_description", "rating"]].copy()
```

In [11]:
```
df.isnull().sum()
```

Out[11]:
```
review_description    0
rating                0
dtype: int64
```

In [17]:
```
df.loc[df["rating"] <= 2, "sentiment"] = "Negative"
df.loc[df["rating"] == 3, "sentiment"] = "Neutral"
df.loc[df["rating"] >= 4, "sentiment"] = "Positive"
```

In [19]:
```
df["sentiment"].value_counts()
```

Out[19]:
```
sentiment
Negative    110949
Positive     81110
Neutral       8732
Name: count, dtype: int64
```

In [21]:
```
df["review_description"] = df["review_description"].str.lower()
df["review_description"] = df["review_description"].str.replace(r'[^a-zA-Z0-9\s]', ''
df["review_description"] = df["review_description"].str.replace(r'\s+', ' ').str.stri
```

In [23]:
```
X = df["review_description"]
Y = df["sentiment"]
```

In [25]:
```
from sklearn.feature_extraction.text import TfidfVectorizer
vectorizer = TfidfVectorizer()

X_vec = vectorizer.fit_transform(X)
```

In [26]:
```
from sklearn.model_selection import train_test_split

x_train, x_test, y_train, y_test = train_test_split(X_vec, Y, test_size=0.2)
```

In [29]:
```
from sklearn.linear_model import LogisticRegression
LR = LogisticRegression()
LR.fit(x_train, y_train)
```

Out[29]:

▾ LogisticRegression ⓘ ?

```
LogisticRegression()
```

In [30]:
```
y_pred = LR.predict(x_test)
```

In [33]:
```
from sklearn.metrics import accuracy_score, classification_report
LR_ACC = accuracy_score(y_test, y_pred)
LR_CR = classification_report(y_test, y_pred)

print("Accuracy :", LR_ACC)
print("Classification Report :\n", LR_CR)
```

```
Accuracy : 0.9105804427401081
Classification Report :
               precision    recall  f1-score   support

    Negative       0.91      0.97      0.94     22187
     Neutral       0.36      0.03      0.06      1741
    Positive       0.92      0.93      0.92     16231

    accuracy                           0.91     40159
   macro avg       0.73      0.64      0.64     40159
weighted avg       0.89      0.91      0.89     40159
```