```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score
import re
import nltk
from nltk import word_tokenize
nltk.download('stopwords')
nltk.download('punkt')
```

```
[nltk_data] Downloading package stopwords to /root/nltk_data...
[nltk_data]   Package stopwords is already up-to-date!
[nltk_data] Downloading package punkt to /root/nltk_data...
[nltk_data]   Package punkt is already up-to-date!
True
```

```
validataion = pd.read_csv('twitter_validation.csv')
train = pd.read_csv('twitter_training.csv')
```

```
train.head()
```

	2401	Borderlands	Positive	im getting on borderlands and i will murder you all ,
0	2401	Borderlands	Positive	I am coming to the borders and I will kill you...
1	2401	Borderlands	Positive	im getting on borderlands and i will kill you ...
2	2401	Borderlands	Positive	im coming on borderlands and i will murder you...
3	2401	Borderlands	Positive	im getting on borderlands 2 and i will murder ...
4	2401	Borderlands	Positive	im getting into borderlands and i can murder y...

Next steps: Generate code with train | View recommended plots | New interactive sheet

```
train.columns = ['id','information','type','text']
validataion.columns = ['id','information','type','text']
```

```
train.head()
```

	id	information	type	text
0	2401	Borderlands	Positive	I am coming to the borders and I will kill you...
1	2401	Borderlands	Positive	im getting on borderlands and i will kill you ...
2	2401	Borderlands	Positive	im coming on borderlands and i will murder you...
3	2401	Borderlands	Positive	im getting on borderlands 2 and i will murder ...
4	2401	Borderlands	Positive	im getting into borderlands and i can murder y...

Next steps: Generate code with train | View recommended plots | New interactive sheet

```
validataion.head()
```

	id	information	type	text
0	352	Amazon	Neutral	BBC News - Amazon boss Jeff Bezos rejects clai...
1	8312	Microsoft	Negative	@Microsoft Why do I pay for WORD when it funct...
2	4371	CS-GO	Negative	CSGO matchmaking is so full of closet hacking,...
3	4433	Google	Neutral	Now the President is slapping Americans in the...
4	6273	FIFA	Negative	Hi @EAHelp I've had Madeleine McCann in my cel...

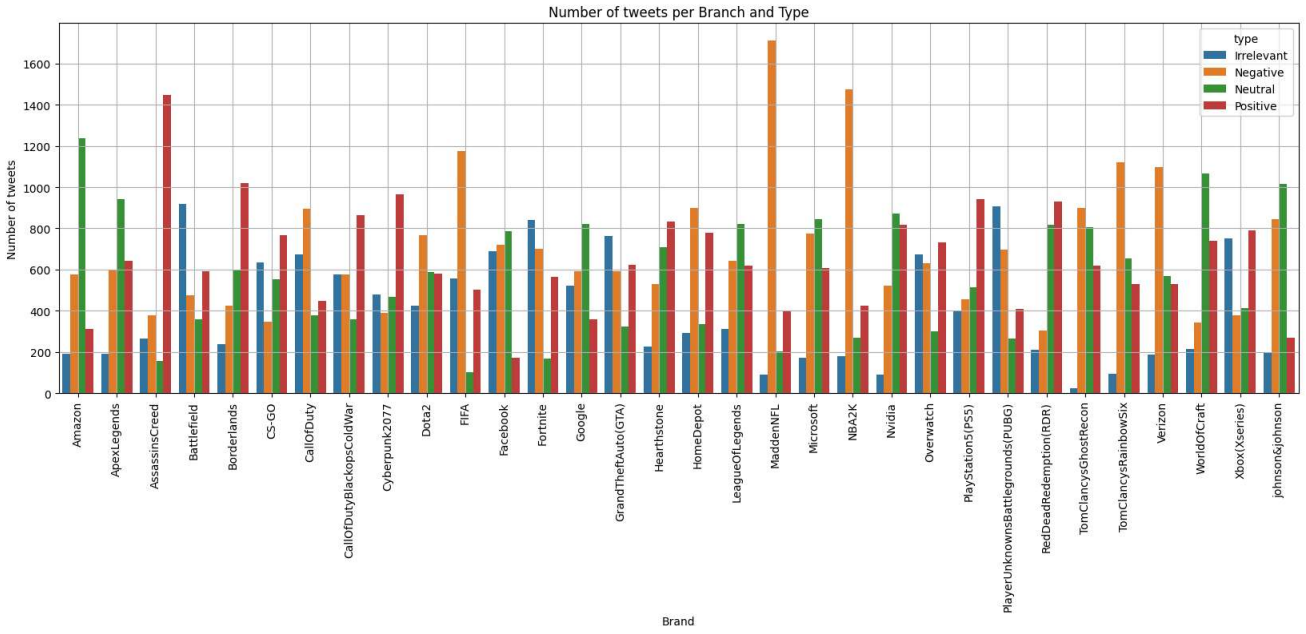
Next steps: Generate code with validataion | View recommended plots | New interactive sheet

```
train_df = train
val_df = validataion


train_df['lower']=train_df.text.str.lower()
train_df['lower']=[str(data) for data in train_df.lower]
train_df['lower']=train_df.lower.apply(lambda x: re.sub('[^A-za-z0-9 ]+','',x))


val_df['lower']=val_df.text.str.lower()
val_df['lower']=[str(data) for data in val_df.lower]
val_df['lower']=val_df.lower.apply(lambda x: re.sub('[^A-za-z0-9 ]+','',x))


train_df.head()
```

	id	information	type	text	lower
0	2401	Borderlands	Positive	I am coming to the borders and I will kill you...	i am coming to the borders and i will kill you...
1	2401	Borderlands	Positive	im getting on borderlands and i will kill you ...	im getting on borderlands and i will kill you all
2	2401	Borderlands	Positive	im coming on borderlands and i will murder you...	im coming on borderlands and i will murder you...
				im getting on borderlands 2	im getting on borderlands 2

Next steps: Generate code with train_df | View recommended plots | New interactive sheet

```
info_type=train.groupby(['information','type']).count().reset_index()
plt.figure(figsize=(20,6))
sns.barplot(x='information',y='id',hue='type',data=info_type)
plt.xticks(rotation=90)
plt.xlabel('Brand')
plt.ylabel('Number of tweets')
plt.grid()
plt.title('Number of tweets per Branch and Type');
```




```
# text splitting
tokens_text =[word_tokenize(str(word)) for word in train_df.lower ]
```

```
tokens_text
```

```
 'itself',
 'is',
 'already',
 'very',
 'brutally',
 'badly',
 'underpaid',
 'for',
 'what',
 'little',
 'we',
 'do',
 'when',
 'vs',
 'others',
 'this',
 'is',
 'shameful'],
['making',
 'games',
 'is',
 'a',
 'particularly',
 'difficult',
 'business',
 'without',
 'the',
 'original',
 'you',
 'have',
 'nothing',
 'our',
 'industry',
 'seems',
 'already',
 'brutally',
 'underpaid',
 'for',
 'what',
 'we',
 'do',
 'around',
 'nintendo',
 'this',
 'is',
 'shameful'],
['sweet'],
['sweet'],
['the', 'sweat'],
['sweet'],
['for', 'sweet'],
['of'],
['whos', 'down', 'for', 'some', 'borderlands', 'on'],
['whos', 'on', 'for', 'some', 'borderlands'],
['whos', 'at', 'borderlands'],
['whos', 'down', 'with', 'some', 'borderlands', 'on'],
['whot', 's', 'someone', 'down', 'for', 'some', 'borderlands', 'on'],
...]
```

```
stopwords_nltk = nltk.corpus.stopwords
stop_words = stopwords_nltk.words('english')
print(stop_words)
```

```
['i', 'me', 'my', 'myself', 'we', 'our', 'ours', 'ourselves', 'you', "you're", "you'v
```

```
bow_counts = CountVectorizer(tokenizer = word_tokenize, stop_words=stop_words, ngram_rang


reviews_train,reviews_test = train_test_split(train_df,test_size=0.2,random_state=42)


x_train_bow = bow_counts.fit_transform(reviews_train.lower)
x_test_bow = bow_counts.transform(reviews_test.lower)
```

```
/usr/local/lib/python3.10/dist-packages/sklearn/feature_extraction/text.py:521: UserW
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/feature_extraction/text.py:406: UserW
  warnings.warn(
```

```
y_train_bow = reviews_train['type']
y_test_bow = reviews_test['type']


y_test_bow
```

	type
34877	Irrelevant
21704	Positive
47008	Negative
7969	Irrelevant
454	Positive
...	...
52360	Irrelevant
57296	Positive
35884	Positive
59060	Negative
4740	Neutral

14937 rows × 1 columns

dtype: object

```
logistic_reg = LogisticRegression(C=1, solver='liblinear',max_iter = 200)
logistic_reg.fit(x_train_bow,y_train_bow)
test_pred = logistic_reg.predict(x_test_bow)
print("Accuracy: ",accuracy_score(y_test_bow,test_pred))
```

```
Accuracy:  0.7954743255004352
/usr/local/lib/python3.10/dist-packages/sklearn/svm/_base.py:1235: ConvergenceWarning
  warnings.warn(
```

```
x_val_bow = bow_counts.transform(val_df.lower)
y_val_bow = val_df['type']
```

```
y_val_bow
```

	type
0	Neutral
1	Negative
2	Negative
3	Neutral
4	Negative
...	...
994	Irrelevant
995	Irrelevant
996	Positive
997	Positive
998	Neutral

999 rows × 1 columns

dtype: object

```
print(x_val_bow)
```

```
  (0, 2414)     1
  (0, 3040)     1
  (0, 4650)     1
  (0, 4654)     1
  (0, 4958)     1
  (0, 5747)     1
  (0, 7352)     1
  (0, 7879)     1
  (0, 9291)     1
  (0, 10643)    1
  (0, 18146)    1
  (0, 19645)    1
  (0, 22447)    1
```

```
(0, 27769)    1
(1, 7254)     1
(1, 13530)    1
(1, 21186)    1
(1, 24254)    1
(1, 25973)    1
(1, 29041)    1
(1, 36790)    1
(2, 4234)     1
(2, 7496)     1
(2, 8754)     1
(2, 13512)    1
:     :
(996, 15278)  1
(996, 22105)  1
(996, 25710)  1
(996, 31737)  1
(996, 31808)  1
(996, 33201)  1
(996, 33375)  1
(996, 33424)  1
(996, 36575)  1
(997, 5779)   1
(997, 13242)  1
(997, 21186)  1
(997, 30451)  1
(997, 33375)  1
(997, 36593)  1
(998, 4347)   1
(998, 6523)   1
(998, 16256)  1
(998, 18314)  2
(998, 26093)  1
(998, 28188)  1
(998, 29522)  1
(998, 31435)  1
(998, 32194)  1
(998, 35185)  1
```

```
val_res = logistic_reg.predict(x_val_bow)
print("Accuracy: ",accuracy_score(y_val_bow,val_res))
```

```
Accuracy:  0.9029029029029029
```

Start coding or generate with AI.