Import Libraries

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

Import Dataset

```
df=pd.read_csv("/content/twitter_validation.csv")
```

```
df.head()
```

	3364	Facebook	Irrelevant	I mentioned on Facebook that I was struggling for motivation to go for a run the other day, which has been translated by Tom’s great auntie as ‘Hayley can’t get out of bed’ and told to his grandma, who now thinks I’m a lazy, terrible person 🤣
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 df View recommended plots

```
df.rename(columns={'3364':'id','Facebook':'app name','Irrelevant':'sentiment','I mentioned on Facebook that I was struggling for motivation t
```

```
df.head()
```

	id	app name	sentiment	Review
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 df View recommended plots

```
df['Review'] = df['Review'].astype(str).apply(lambda x: x.lower())
df['Review'] = df['Review'].apply(lambda x: x.replace(r'[^\w\s]', ''))
```

```
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score, classification_report
```

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

```
vectorizer = TfidfVectorizer()
```

Fit the vectorizer

```
X_train_tfidf = vectorizer.fit_transform(X_train)
X_test_tfidf = vectorizer.transform(X_test)
```

Train model

```
model = LogisticRegression()
```

```
model.fit(X_train_tfidf, y_train)
```

```
▾ LogisticRegression
LogisticRegression()
```

Evaluate model

```
y_pred = model.predict(X_test_tfidf)
print('Accuracy:', accuracy_score(y_test, y_pred))
```

```
Accuracy: 0.475
```

```
print('Classification Report:')
print(classification_report(y_test, y_pred))
```

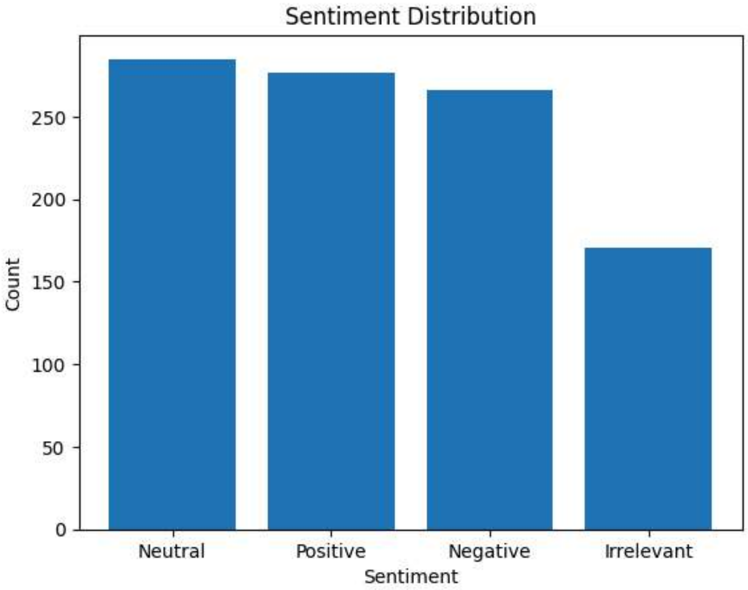
```
Classification Report:
              precision    recall  f1-score   support

  Irrelevant       0.44      0.11      0.17        37
    Negative       0.49      0.69      0.57        52
     Neutral       0.55      0.46      0.50        59
    Positive       0.41      0.54      0.47        52

    accuracy                           0.48       200
   macro avg       0.47      0.45      0.43       200
weighted avg       0.48      0.47      0.45       200
```

Visualizations

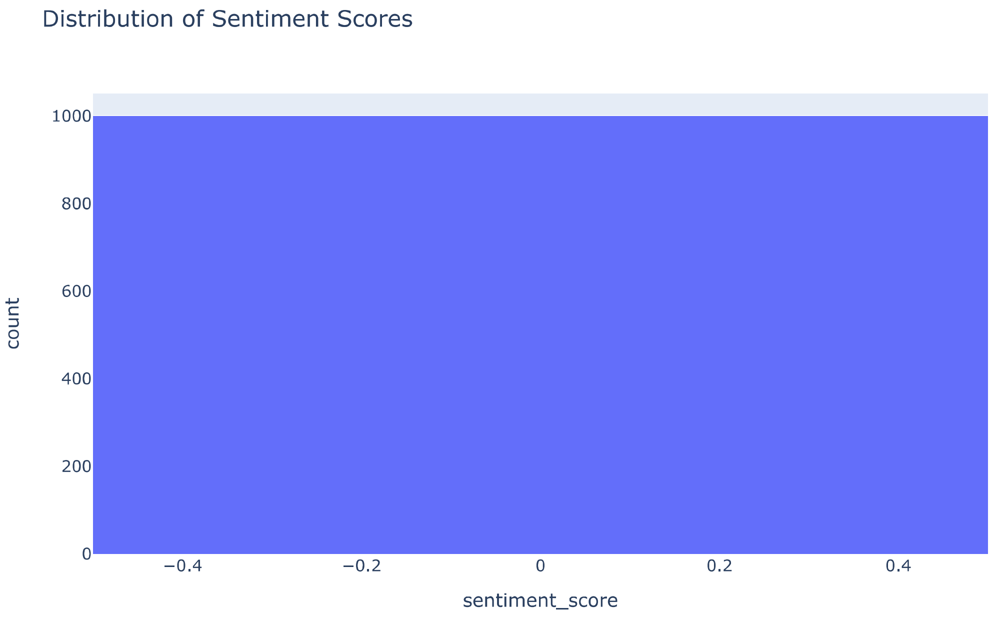
```
df['sentiment_score'] = df['sentiment'].apply(lambda x: 1 if x == 'positive' else -1 if x == 'negative' else 0)
```

Bar Graph

```
sentiment_counts = df['sentiment'].value_counts()
plt.bar(sentiment_counts.index, sentiment_counts.values)
plt.xlabel('Sentiment')
plt.ylabel('Count')
plt.title('Sentiment Distribution')
plt.show()
```

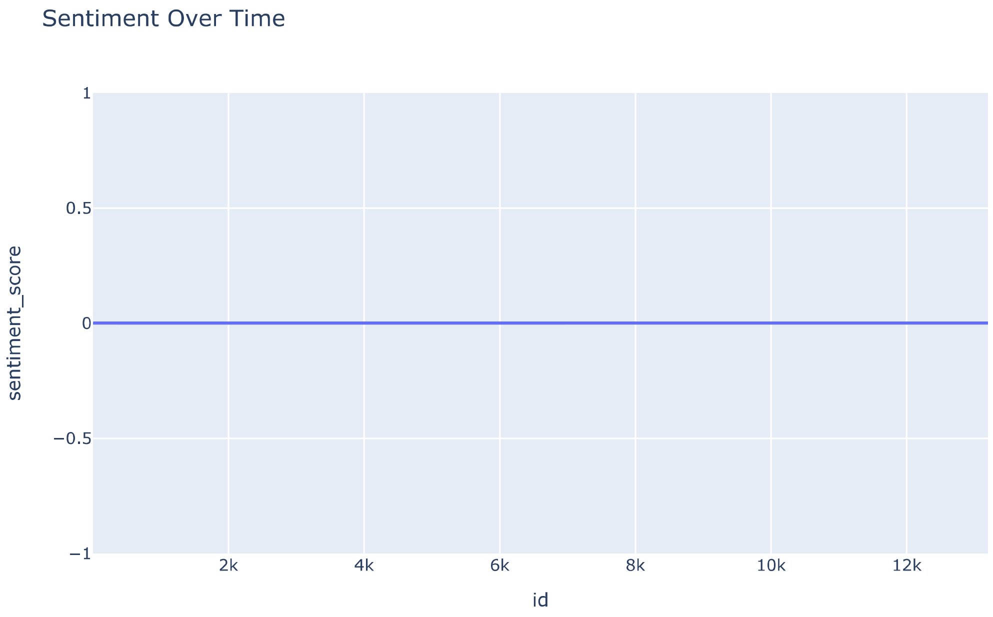


```
import plotly.express as px

fig = px.histogram(df, x="sentiment_score", title="Distribution of Sentiment Scores")
fig.show()
```



```
import plotly.express as px
print(df.columns)
fig = px.line(df, x="id", y="sentiment_score", title="Sentiment Over Time")
fig.show()
```

```
Index(['id', 'app name', 'sentiment', 'Review', 'sentiment_score'], dtype='object')
```



Test the model

```
y_test
```

```
453     Neutral
793     Neutral
209     Neutral
309    Negative
740    Negative
         ...
78      Neutral
29     Positive
```

```
277    Positive
261     Neutral
423     Neutral
Name: sentiment, Length: 200, dtype: object
```

```
y_pred
```

```
array(['Neutral', 'Positive', 'Neutral', 'Negative', 'Negative',
       'Neutral', 'Negative', 'Positive', 'Irrelevant', 'Negative',
       'Negative', 'Negative', 'Neutral', 'Positive', 'Positive',
       'Neutral', 'Negative', 'Positive', 'Negative', 'Positive',
       'Positive', 'Positive', 'Neutral', 'Negative', 'Negative',
       'Negative', 'Negative', 'Positive', 'Negative', 'Positive',
       'Neutral', 'Negative', 'Neutral', 'Negative', 'Negative',
       'Negative', 'Neutral', 'Positive', 'Positive', 'Positive',
       'Neutral', 'Neutral', 'Negative', 'Neutral', 'Neutral', 'Neutral',
       'Negative', 'Positive', 'Positive', 'Neutral', 'Irrelevant',
       'Positive', 'Positive', 'Negative', 'Positive', 'Positive',
       'Positive', 'Neutral', 'Negative', 'Positive', 'Neutral',
       'Neutral', 'Negative', 'Irrelevant', 'Positive', 'Positive',
       'Positive', 'Negative', 'Positive', 'Neutral', 'Neutral',
       'Positive', 'Negative', 'Neutral', 'Neutral', 'Positive',
       'Negative', 'Neutral', 'Neutral', 'Neutral', 'Positive',
       'Negative', 'Positive', 'Negative', 'Negative', 'Neutral',
       'Positive', 'Negative', 'Negative', 'Irrelevant', 'Positive',
       'Negative', 'Negative', 'Negative', 'Negative', 'Positive',
       'Negative', 'Positive', 'Neutral', 'Negative', 'Neutral',
       'Negative', 'Positive', 'Neutral', 'Positive', 'Neutral',
       'Positive', 'Negative', 'Positive', 'Positive', 'Neutral',
       'Negative', 'Negative', 'Negative', 'Positive', 'Irrelevant',
       'Negative', 'Negative', 'Negative', 'Negative', 'Positive',
       'Neutral', 'Positive', 'Negative', 'Positive', 'Positive',
       'Irrelevant', 'Positive', 'Neutral', 'Negative', 'Negative',
       'Positive', 'Negative', 'Negative', 'Negative', 'Negative',
       'Negative', 'Positive', 'Positive', 'Positive', 'Negative',
       'Negative', 'Negative', 'Negative', 'Neutral', 'Negative',
       'Neutral', 'Positive', 'Negative', 'Neutral', 'Positive',
       'Negative', 'Negative', 'Positive', 'Negative', 'Negative',
       'Neutral', 'Positive', 'Positive', 'Positive', 'Neutral',
       'Neutral', 'Positive', 'Neutral', 'Negative', 'Positive',
       'Negative', 'Positive', 'Neutral', 'Negative', 'Irrelevant',
       'Negative', 'Positive', 'Positive', 'Negative', 'Negative',
       'Positive', 'Neutral', 'Neutral', 'Negative', 'Positive',
       'Neutral', 'Positive', 'Neutral', 'Neutral', 'Irrelevant',
       'Positive', 'Neutral', 'Negative', 'Neutral', 'Positive',
       'Negative', 'Irrelevant', 'Positive', 'Negative', 'Positive',
       'Negative', 'Positive', 'Neutral', 'Positive'], dtype=object)
```