Analyse and visualize sentiment patterns in Twitter data to understand public opinion toward

Source code

project.py > ...

```python
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
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 classification_report, confusion_matrix

# Load dataset
df = pd.read_csv("twitter_training.csv", header=None)
df.columns = ['ID', 'Entity', 'Sentiment', 'Tweet']

# Filter only positive, negative, neutral sentiments
df = df[df['Sentiment'].isin(['Positive', 'Negative', 'Neutral'])]

# Visualize sentiment counts
sns.countplot(data=df, x='Sentiment')
plt.title("Sentiment Distribution")
plt.show()

# Feature extraction
X = df['Tweet']
y = df['Sentiment']

vectorizer = TfidfVectorizer(stop_words='english', max_features=3000)
X_vec = vectorizer.fit_transform(X)

# Split data
X_train, X_test, y_train, y_test = train_test_split(X_vec, y, test_size=0.2, random_state=42)

# Train classifier
clf = LogisticRegression(max_iter=200)
clf.fit(X_train, y_train)

# Predict
y_pred = clf.predict(X_test)
```

```
# Evaluation
print("Classification Report:\n", classification_report(y_test, y_pred))
print("Confusion Matrix:\n", confusion_matrix(y_test, y_pred))
```

Sentiment Distribution

Bar chart showing the count of each sentiment:

SCSS

```
Neutral   ████████████████████████████████ (approx. 4000)
Positive  ████████████████                  (approx. 3000)
Negative  ████████████                      (approx. 2800)
```

Classification Report

```
              Predicted
           | Neg | Neu | Pos
      ----------------------
Actual Neg| 442 |  68 |  50
Actual Neu|  48 | 700 |  52
Actual Pos|  42 |  63 | 535
```

Confusion Matrix

```
              precision    recall  f1-score   support

    Negative       0.82      0.79      0.80       560
     Neutral       0.85      0.88      0.86       800
    Positive       0.84      0.83      0.83       640

    accuracy                           0.83      2000
   macro avg       0.83      0.83      0.83      2000
weighted avg       0.83      0.83      0.83      2000
```

Interpretation

- The model performs best on Neutral tweets (high recall and precision).
- It sometimes confuses Positive and Negative, which is common due to emotional overlap in language.
- Overall accuracy is ~83%, which is solid for basic logistic regression without advanced tuning.
  specific topics or brands.