	precision	recall	f1-score	support
negative	0.52	0.35	0.42	1562
neutral	0.49	0.76	0.60	2230
positive	0.68	0.40	0.51	1705
accuracy			0.53	5497
macro avg	0.57	0.51	0.51	5497
weighted avg	0.56	0.53	0.52	5497

```
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.neighbors import KNeighborsClassifier
from sklearn.metrics import classification_report

# Load the dataset
# Please replace 'dataset.csv' with your actual dataset link
df = pd.read_csv('G:/Games/install/dataset.csv',encoding='unicode_escape')

# Fill NaN values with an empty string
df['text'] = df['text'].fillna('')

# Split the dataset into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(df['text'], df['sentiment'], test_size=0.2,
random_state=42)

# Vectorize the text data
vectorizer = CountVectorizer()
X_train_vec = vectorizer.fit_transform(X_train)
X_test_vec = vectorizer.transform(X_test)

# Train the KNN classifier
knn = KNeighborsClassifier(n_neighbors=3)
knn.fit(X_train_vec, y_train)

# Predict the sentiment for the testing set
y_pred = knn.predict(X_test_vec)

# Print the classification report
print(classification_report(y_test, y_pred))
```