In [3]:
```python
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import re
import string
```

In [4]:
```python
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 accuracy_score, classification_report, confusion_matrix
```

In [5]:
```python
df = pd.read_csv("IMDB Dataset.csv")
```

In [6]:
```python
df.head()
```

Out[6]:

	review	sentiment
0	One of the other reviewers has mentioned that ...	positive
1	A wonderful little production. <br /><br />The...	positive
2	I thought this was a wonderful way to spend ti...	positive
3	Basically there's a family where a little boy ...	negative
4	Petter Mattei's "Love in the Time of Money" is...	positive

In [7]:
```python
df['sentiment'] = df['sentiment'].map({'positive': 1, 'negative': 0})
print(df['sentiment'].value_counts())
```

```
sentiment
1    25000
0    25000
Name: count, dtype: int64
```

In [8]:
```python
def clean_text(text):
    text = text.lower()
    text = re.sub(f"[{re.escape(string.punctuation)}]", "", text)
    text = re.sub(r"\d+", "", text)
    text = re.sub(r"\s+", " ", text)
    return text.strip()

df['review'] = df['review'].apply(clean_text)
df.head()
```

Out[8]:

	review	sentiment
0	one of the other reviewers has mentioned that ...	1
1	a wonderful little production br br the filmin...	1
2	i thought this was a wonderful way to spend ti...	1
3	basically theres a family where a little boy j...	0
4	petter matteis love in the time of money is a ...	1

In [9]:
```python
X = df['review']
y = df['sentiment']
# Split into training and testing data
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
```

In [14]:
```python
vectorizer = TfidfVectorizer(max_features=5000)
X_train_tfidf = vectorizer.fit_transform(X_train)
X_test_tfidf = vectorizer.transform(X_test)
```

In [19]:
```python
model = LogisticRegression()
model.fit(X_train_tfidf, y_train)
```

Out[19]:
```
▾ LogisticRegression
LogisticRegression()
```

In [18]:
```python
y_pred = model.predict(X_test_tfidf)
# Accuracy
accuracy = accuracy_score(y_test, y_pred)
print("Accuracy:", accuracy)
# Detailed report
print("\nClassification Report:\n", classification_report(y_test, y_pred))
```

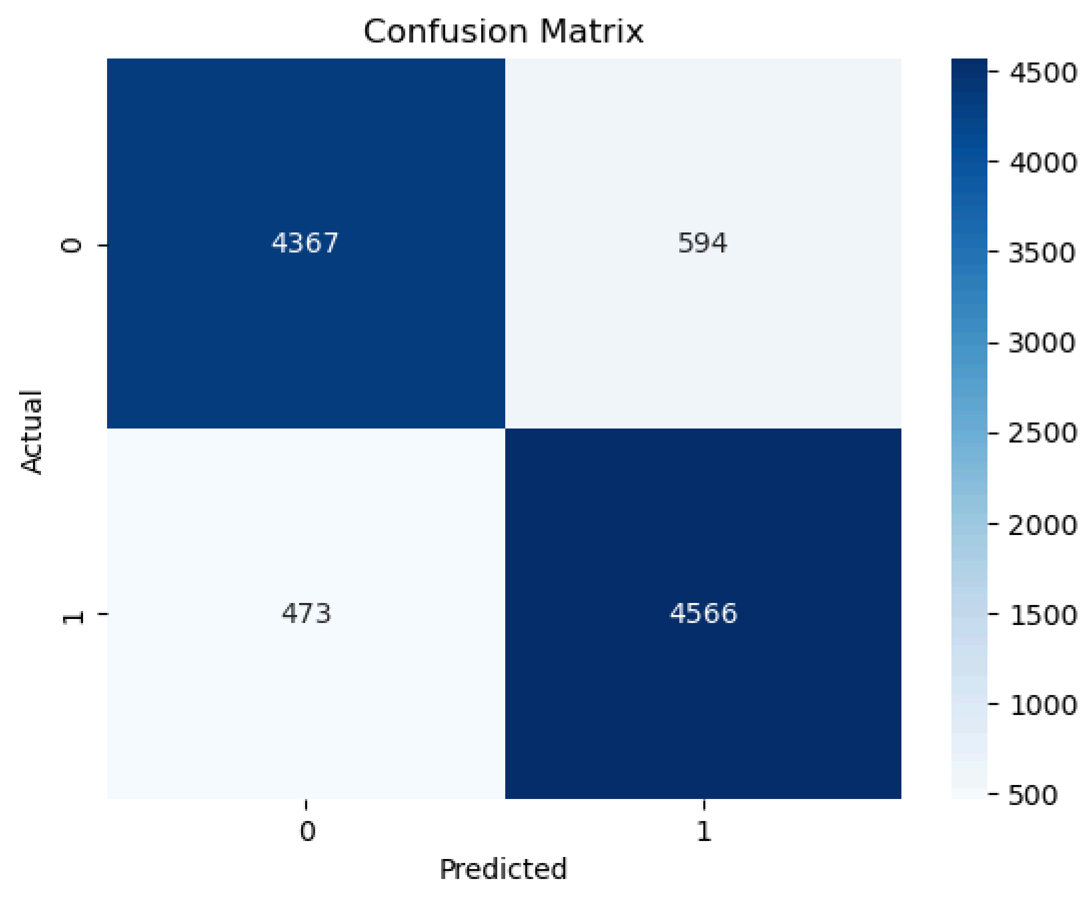
```
Accuracy: 0.8933

Classification Report:
               precision    recall  f1-score   support

           0       0.90      0.88      0.89      4961
           1       0.88      0.91      0.90      5039

    accuracy                           0.89     10000
   macro avg       0.89      0.89      0.89     10000
weighted avg       0.89      0.89      0.89     10000
```

In [13]:
```python
# Confusion Matrix
conf_mat = confusion_matrix(y_test, y_pred)
sns.heatmap(conf_mat, annot=True, fmt='d', cmap='Blues')
plt.xlabel("Predicted")
plt.ylabel("Actual")
plt.title("Confusion Matrix")
plt.show()
```



In [ ]: