TASK -4

SENTIMENT ANALYSIS

Data Source: https://www.kaggle.com/datasets/crowdflower/twitter-airlinesentiment

Jupyter Notebook: Sentiment Analysis with NLP

1. Import Libraries

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

from sklearn.model_selection import train_test_split
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.naive_bayes import MultinomialNB
from sklearn.metrics import classification_report, accuracy_score
```

2. Load and Explore Dataset

```python
df = pd.read_csv('sentiment_data.csv')
print(df.head())
print(df['sentiment'].value_counts())
```

3. Preprocessing Text

python

```python
def clean_text(text):
    text = text.lower()
    text = re.sub(r"[^a-zA-Z\s]", "", text)
    return text

df['clean_text'] = df['text'].apply(clean_text)
```

4. Convert Text to Vectors (Bag of Words)

python

```python
vectorizer = CountVectorizer(stop_words='english')
X = vectorizer.fit_transform(df['clean_text'])
y = df['sentiment']
```

5. Split Dataset

python

```python
X_train, X_test, y_train, y_test = train_test_split(X, y,
```

```
t(X, y, test_size=0.2, random_state=42)
```

6. Train Sentiment Model (Naive Bayes)

python

```python
model = MultinomialNB()
model.fit(X_train, y_train)
y_pred = model.predict(X_test)
```

7. Evaluation

python

```python
print("Accuracy:", accuracy_score(y_test, y_pred))
print("Report:\n", classification_report(y_test, y_pred))
```

8. Try Custom Input

python

```python
def predict_sentiment(text):
    text = clean_text(text)
    vec = vectorizer.transform([text])
    return model.predict(vec)[0]

print(predict_sentiment("I hate this service."))
print(predict_sentiment("Amazing quality and speed."))
```

Sample Output:

markdown

```
Accuracy: 1.0
              precision    recall  f1-score   support

     neutral       1.00      1.00      1.00         1
     negative      1.00      1.00      1.00         1

     positive       1.00      1.00      1.00         1
```