Experiment 12 Classification of Facebook data

To perform classification on Facebook data in Python, you would typically need a dataset with labeled examples (e.g., posts labeled as positive, negative, or neutral). Once you have a labeled dataset, you can use various machine learning algorithms for classification. Here's a step-by-step guide using the scikit-learn library, one of the most popular machine learning libraries in Python:

Step 1: Import Libraries and Load Data

```
import pandas as pd

from sklearn.feature_extraction.text import CountVectorizer

from sklearn.model_selection import train_test_split

from sklearn.naive_bayes import MultinomialNB

from sklearn.metrics import accuracy_score, classification_report

# Load your Facebook data into a pandas DataFrame (assuming you have 'text' and 'label' columns)

df = pd.read_csv('your_facebook_data.csv')
```

Step 2: Preprocess the Data

```
# Convert text data into numerical features using CountVectorizer

vectorizer = CountVectorizer()

X = vectorizer.fit_transform(df['text'])

# Split the data into training and testing sets

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

Step 3: Train a Classifier (Naive Bayes in this example)

```
# Initialize and train a Naive Bayes classifier

classifier = MultinomialNB()

classifier.fit(X_train, y_train)
```

Step 4: Make Predictions and Evaluate the Model

```
# Make predictions on the test set

predictions = classifier.predict(X_test)

# Calculate accuracy and other metrics

accuracy = accuracy_score(y_test, predictions)
```

```
report = classification_report(y_test, predictions)

print(f'Accuracy: {accuracy:.2f}')

print('Classification Report:\n', report)
```

we used the Naive Bayes classifier, but you can use other classifiers available in scikit-learn (such as RandomForestClassifier, SVM, etc.) based on the nature of your data and problem.