Лабораторна робота №1 – Product Reviews

```
import nltk
import numpy as np
import re
import string
from nltk.corpus import twitter_samples, stopwords
from nltk.stem import PorterStemmer
from nltk.tokenize import TweetTokenizer
```

ЗАВДАННЯ 2: Попередня обробка тексту

```
def process_tweet(tweet):
    """
    Токенізація, видалення стоп-слів, стемінг
    """
    stemmer = PorterStemmer()
    stopwords_english = stopwords.words('english')

    tweet = re.sub(r'\$\w*', '', tweet)
    tweet = re.sub(r'^RT[\s]+', '', tweet)
    tweet = re.sub(r'https?://[^\s\n\r]+', '', tweet)
    tweet = re.sub(r'#', '', tweet)

    tokenizer = TweetTokenizer(
        preserve_case=False,
        strip_handles=True,
        reduce_len=True
    )

    tweet_tokens = tokenizer.tokenize(tweet)

    tweets_clean = []
    for word in tweet_tokens:
        if word not in stopwords_english and word not in string.punctuation:
            tweets_clean.append(stemmer.stem(word))

    return tweets_clean
```

ЗАВДАННЯ 3: Побудова словника частотності

```
def build_freqs(tweets, ys):
    """
    freqs[(word, label)] = count
    """
    ys = np.squeeze(ys).tolist()
    freqs = {}

    for y, tweet in zip(ys, tweets):
        for word in process_tweet(tweet):
            freqs[(word, y)] = freqs.get((word, y), 0) + 1

    return freqs
```

ЗАВДАННЯ 4: Логістична регресія

```
def sigmoid(z):
    return 1 / (1 + np.exp(-z))


def gradientDescent(x, y, theta, alpha, num_iters):
    m = x.shape[0]
    y = y.reshape(-1, 1)

    for i in range(num_iters):
        z = np.dot(x, theta)
        h = sigmoid(z)

        epsilon = 1e-15
        h = np.clip(h, epsilon, 1 - epsilon)

        J = (-1 / m) * np.sum(y * np.log(h) + (1 - y) * np.log(1 - h))
        grad = (1 / m) * np.dot(x.T, (h - y))

        theta = theta - alpha * grad

        if i % 200 == 0:
            print(f"Ітерація {i}, J = {J:.6f}")

    return J, theta
```

ЗАВДАННЯ 5: Ознаки та передбачення

```
def extract_features(tweet, freqs):
    """
    [bias, positive_freq, negative_freq]
    """
    words = process_tweet(tweet)
    x = np.zeros((1, 3))
    x[0, 0] = 1

    for word in words:
        x[0, 1] += freqs.get((word, 1.0), 0)
        x[0, 2] += freqs.get((word, 0.0), 0)

    return x


def predict_tweet(tweet, freqs, theta):
    x = extract_features(tweet, freqs)
    return sigmoid(np.dot(x, theta))


def test_logistic_regression(test_x, test_y, freqs, theta):
    y_hat = []

    for tweet in test_x:
        y_pred = predict_tweet(tweet, freqs, theta)
        y_hat.append(1.0 if y_pred > 0.5 else 0.0)

    return np.mean(np.array(y_hat) == np.squeeze(test_y))
```

ГОЛОВНИЙ БЛОК

```
if __name__ == "__main__":

    nltk.download('twitter_samples')
```

```
nltk.download('twitter_samples')
nltk.download('stopwords')

print("Завантаження корпусу...")

pos = twitter_samples.strings('positive_tweets.json')
neg = twitter_samples.strings('negative_tweets.json')

train_pos = pos[:4000]
test_pos = pos[4000:]
train_neg = neg[:4000]
test_neg = neg[4000:]

train_x = train_pos + train_neg
test_x = test_pos + test_neg

train_y = np.append(np.ones(len(train_pos)), np.zeros(len(train_neg)))
test_y = np.append(np.ones(len(test_pos)), np.zeros(len(test_neg)))

print("Побудова словника...")
freqs = build_freqs(train_x, train_y)

print("Навчання моделі...")

X = np.zeros((len(train_x), 3))
for i in range(len(train_x)):
    X[i, :] = extract_features(train_x[i], freqs)

alpha = 1e-9
num_iters = 1500

J, theta = gradientDescent(X, train_y, np.zeros((3, 1)), alpha, num_iters)

print("\nНавчання завершено")
print("Втрати:", J)
print("Theta:", np.squeeze(theta))

accuracy = test_logistic_regression(test_x, test_y, freqs, theta)
print(f"\nТочність: {accuracy * 100:.2f}%")

print("\nТестування власного твіту:")
tweet = "I am very happy and excited about this project"
pred = predict_tweet(tweet, freqs, theta)
print(tweet)
print("Позитивний" if pred > 0.5 else "Негативний")
```

```
[nltk_data] Downloading package twitter_samples to /root/nltk_data...
[nltk_data]   Unzipping corpora/twitter_samples.zip.
[nltk_data] Downloading package stopwords to /root/nltk_data...
[nltk_data]   Package stopwords is already up-to-date!
Завантаження корпусу...
Побудова словника...
Навчання моделі...
Ітерація 0, J = 0.693147
Ітерація 200, J = 0.522064
Ітерація 400, J = 0.421057
Ітерація 600, J = 0.355746
Ітерація 800, J = 0.310570
Ітерація 1000, J = 0.277694
Ітерація 1200, J = 0.252807
Ітерація 1400, J = 0.233366

Навчання завершено
Втрати: 0.22524410259587288
Theta: [ 5.97380718e-08  5.37857205e-04 -5.58847110e-04]

Точність: 99.65%

Тестування власного твіту:
I am very happy and excited about this project
Позитивний
```