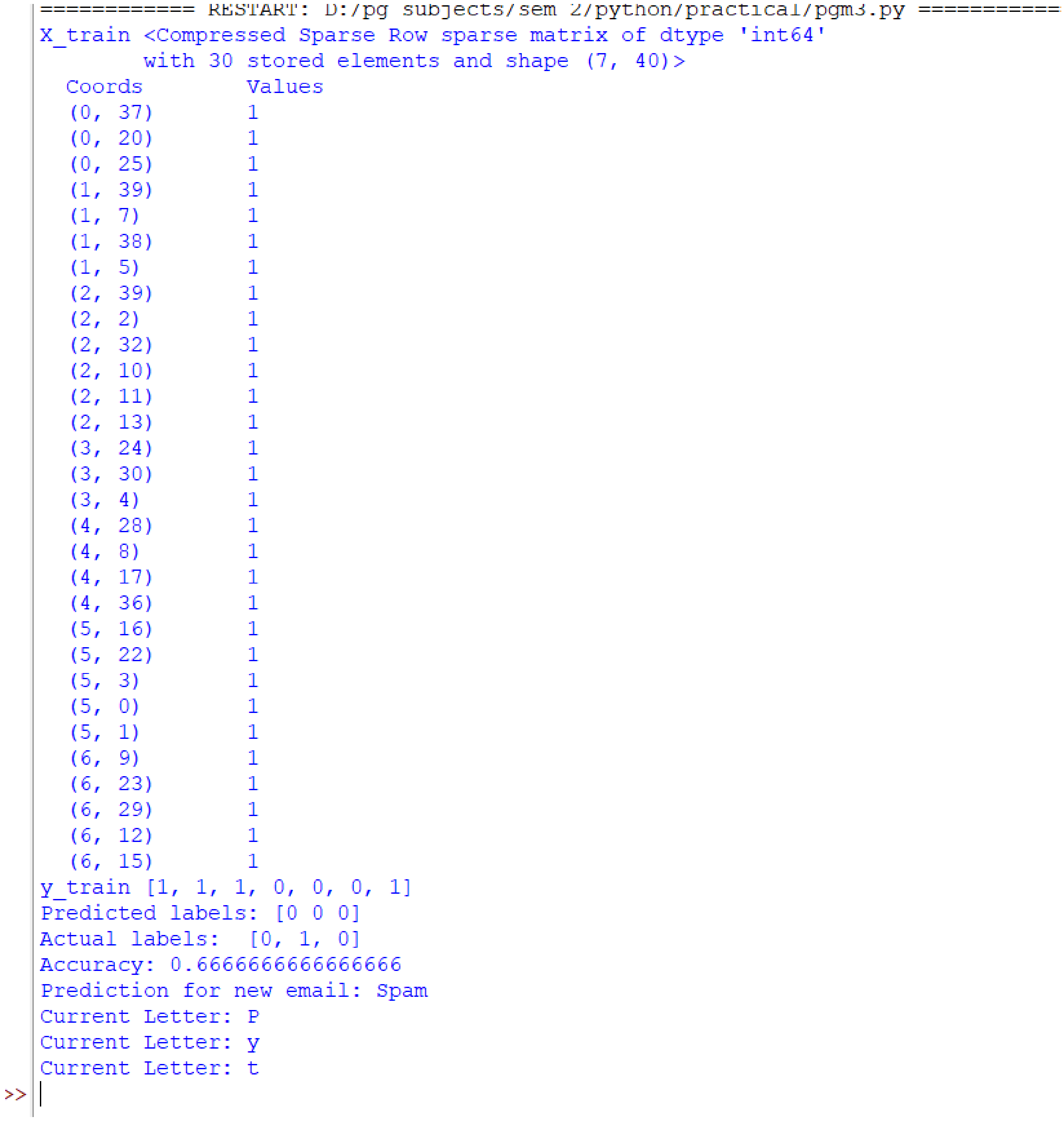
APPLIED ARTIFICIAL INTELLIGENCE

EXPERIMENT – 03

```
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.naive_bayes import MultinomialNB
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score
# 1. Sample dataset (text + labels)
emails = [
    "Win a lottery now", # spam
    "Limited time offer, claim prize", # spam
    "You are selected for a free gift", # spam
    "Important meeting at 10 AM", # not spam
    "Project deadline is tomorrow", # not spam
    "Let’s have lunch today", # not spam
    "Earn money quickly from home", # spam
    "Congratulations, you won a car", # spam
    "Team meeting rescheduled", # not spam
    "Monthly report attached", # not spam
]
labels = [1, 1, 1, 0, 0, 0, 1, 1, 0, 0] # 1 = spam, 0 = not spam
# 2. Convert text into numeric features
vectorizer = CountVectorizer()
X = vectorizer.fit_transform(emails)
# 3. Split into training and testing
X_train, X_test, y_train, y_test = train_test_split(X, labels, test_size=0.3, random_state=42)
print("X_train", X_train)
print("y_train", y_train)
# 4. Train Naïve Bayes classifier
model = MultinomialNB()
model.fit(X_train, y_train)
# 5. Predict on test data
```

```
y_pred = model.predict(X_test)

# 6. Evaluate

print("Predicted labels:", y_pred)

print("Actual labels: ", y_test)

print("Accuracy:", accuracy_score(y_test, y_pred))

# 7. Try a new email

new_email = ["Congratulations! You have won a free iPhone"]

new_data = vectorizer.transform(new_email)

prediction = model.predict(new_data)

print("Prediction for new email:", "Spam" if prediction[0] == 1 else "Not Spam")

for letter in "Python":

    if letter == "h":

        break

    print("Current Letter:", letter)
```



```
============ RESTART: D:/pg subjects/sem 2/python/practical/pgm3.py ===========
X_train <Compressed Sparse Row sparse matrix of dtype 'int64'
        with 30 stored elements and shape (7, 40)>
  Coords        Values
  (0, 37)       1
  (0, 20)       1
  (0, 25)       1
  (1, 39)       1
  (1, 7)        1
  (1, 38)       1
  (1, 5)        1
  (2, 39)       1
  (2, 2)        1
  (2, 32)       1
  (2, 10)       1
  (2, 11)       1
  (2, 13)       1
  (3, 24)       1
  (3, 30)       1
  (3, 4)        1
  (4, 28)       1
  (4, 8)        1
  (4, 17)       1
  (4, 36)       1
  (5, 16)       1
  (5, 22)       1
  (5, 3)        1
  (5, 0)        1
  (5, 1)        1
  (6, 9)        1
  (6, 23)       1
  (6, 29)       1
  (6, 12)       1
  (6, 15)       1
y_train [1, 1, 1, 0, 0, 0, 1]
Predicted labels: [0 0 0]
Actual labels:  [0, 1, 0]
Accuracy: 0.6666666666666666
Prediction for new email: Spam
Current Letter: P
Current Letter: y
Current Letter: t
>>
```