Loading the spam data: The data is loaded from a CSV file into a pandas DataFrame. This step prepares the data for modeling.

Building and training a logistic regression model: The features and target variable are separated, and the data is split into training and testing sets. A logistic regression model is then instantiated, fitted to the training data, and evaluated for accuracy on the test data.

Preparing the first email for prediction: The features of the first email from the emails.txt file are manually extracted and formatted to match the model's input requirements. These features include the number of words, links, capitalized words, and spam words.

Predicting if the first email is spam: The logistic regression model is used to predict whether the first email is classified as spam based on its features.

In [14]:
```
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score

# Load the spam data
spam_data_path = 'spam-data.csv'
spam_data = pd.read_csv(spam_data_path)

# Separating features and target variable
X = spam_data.drop('Class', axis=1)
y = spam_data['Class']

# Splitting the data into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# Building and training the logistic regression model
logistic_model = LogisticRegression(max_iter=1000)
logistic_model.fit(X_train, y_train)

# Evaluating the model
predictions = logistic_model.predict(X_test)
accuracy = accuracy_score(y_test, predictions)

# Loading and preparing the first email for prediction
emails_path = 'emails.txt'
with open(emails_path, 'r') as file:
    emails = file.read().split('----------------')

# Extracting features from the first email manually
# Number of Words, Number of Links, Number of Capitalized Words, Number of Spam Words
email1_features = [[68, 4, 1, 4]]  # Based on the content of the first email in emails.txt

# Predicting if the first email is spam
email1_prediction = logistic_model.predict(email1_features)
if email1_prediction[0] == 1:
    print("The email is predicted to be spam.")
else:
    print("The email is predicted to be not spam.")

print("Accuracy:",accuracy)
```

```
The email is predicted to be spam.
Accuracy: 0.9310344827586207
```

```
/opt/anaconda3/lib/python3.11/site-packages/sklearn/base.py:439: UserWarning: X does not have valid feature names, but LogisticRegression was fitted with feature names
  warnings.warn(
```