```python
# Import necessary libraries
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn.tree import DecisionTreeClassifier
from sklearn.feature_extraction.text import CountVectorizer

# Sample data: features (email content) and labels (0 for not spam, 1 for spam)
emails = [
    "Hey, congratulations! You've won a free cruise. Click here to claim your prize.",
    "Hello, please find the attached document for your review.",
    "URGENT: Your account is about to be suspended. Click the link to verify your details.",
    "Reminder: Don't forget to attend the meeting tomorrow.",
    "Meet hot singles in your area now! Click the link to chat.",
    "Dear customer, your order has been shipped. Click here to track your package."
]
labels = np.array([1, 0, 1, 0, 1, 0])  # 1 for spam, 0 for not spam

# Convert text data into numerical features using CountVectorizer
vectorizer = CountVectorizer()
X = vectorizer.fit_transform(emails)

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

# Create decision tree classifier
clf = DecisionTreeClassifier()

# Train the classifier
clf.fit(X_train, y_train)

# Evaluate the classifier
```

```
accuracy = clf.score(X_test, y_test)
print("Accuracy:", accuracy)


# Example to predict whether a new email is spam or not
new_email = ["Congratulations! You've won a free vacation."]
new_email_features = vectorizer.transform(new_email)
prediction = clf.predict(new_email_features)
if prediction == 1:
    print("This email is predicted to be spam.")
else:
    print("This email is predicted to be not spam.")
```

output:

```
Accuracy: 0.5
This email is predicted to be spam.
```