Python code

```python
import nltk
import string
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

from nltk.corpus import stopwords
from sklearn.model_selection import train_test_split
from sklearn.pipeline import Pipeline
from sklearn.feature_extraction.text import CountVectorizer, TfidfTransformer
from sklearn.naive_bayes import MultinomialNB
from sklearn.metrics import classification_report, confusion_matrix

# Download stopwords
nltk.download('stopwords')

# Load dataset
messages = pd.read_csv('spam.csv', encoding='latin-1')
messages.drop(['Unnamed: 2', 'Unnamed: 3', 'Unnamed: 4'], axis=1, inplace=True)
messages = messages.rename(columns={'v1': 'class', 'v2': 'text'})

# Add message length feature
messages['length'] = messages['text'].apply(len)

# Plot message length distribution
messages.hist(column='length', by='class', bins=50, figsize=(15, 6))
plt.show()

# Function to clean text
def process_text(text):
    """
    1. Remove punctuation
    2. Remove stopwords
    3. Return list of clean words
    """
    nopunc = ''.join([char for char in text if char not in string.punctuation])
    clean_words = [word for word in nopunc.split() if word.lower() not in
stopwords.words('english')]
    return clean_words

# Apply text processing
messages['text'] = messages['text'].apply(process_text)

# Train-test split
msg_train, msg_test, class_train, class_test = train_test_split(
```

```python
    messages['text'], messages['class'], test_size=0.2, random_state=42
)

# Build pipeline
pipeline = Pipeline([
    ('bow', CountVectorizer(analyzer=lambda x: x)),  # Use pre-tokenized input
    ('tfidf', TfidfTransformer()),
    ('classifier', MultinomialNB())
])

# Train the model
pipeline.fit(msg_train, class_train)

# Predict on test data
predictions = pipeline.predict(msg_test)

# Print performance metrics
print(classification_report(class_test, predictions))

# Plot confusion matrix
sns.heatmap(confusion_matrix(class_test, predictions), annot=True, fmt='d')
plt.xlabel("Predicted")
plt.ylabel("Actual")
plt.show()
```