EX.NO.: 01
DATE: 17.06.2025

EMAIL SPAM FILTERING USING ML

AIM

To design and implement a Machine Learning-based Email Spam Filter in Python that classifies emails as 'Spam' or 'Not Spam' using the Naive Bayes classifier, and simulates the deletion of spam emails.

ALGORITHM

1. Download dataset, load it and preprocess the dataset by labeling ‘ham’ as 0 and ‘spam’ as 1
2. Split dataset into training and testing sets
3. Initialize Multinomial Naive Bayes classifier
4. Evaluate model and predict labels of the test data
5. Calculate accuracy, precision, recall, f1 score and confusion matrix
6. Define a function to classify any given email text as ‘Spam’ or ‘Not Spam’

CODE AND OUTPUT

```
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.feature_extraction.text import CountVectorizer

# Load Dataset
url =
"https://raw.githubusercontent.com/justmarkham/pycon-2016-tutorial/master/data/sms.tsv"
data = pd.read_csv(url, sep='\t', header=None, names=['label', 'message'])

print("Dataset Sample:")
print(data.head())

# Convert labels to binary: ham = 0, spam = 1
data['label_num'] = data.label.map({'ham': 0, 'spam': 1})

# Split into train and test sets
X_train, X_test, y_train, y_test = train_test_split(
    data['message'], data['label_num'], test_size=0.2, random_state=42)

# Feature Extraction (Bag of Words)
vectorizer = CountVectorizer()
X_train_vectorized = vectorizer.fit_transform(X_train)
X_test_vectorized = vectorizer.transform(X_test)
```

```
Dataset Sample:
  label                                        message
0   ham  Go until jurong point, crazy.. Available only ...
1   ham                      Ok lar... Joking wif u oni...
2  spam  Free entry in 2 a wkly comp to win FA Cup fina...
3   ham  U dun say so early hor... U c already then say...
4   ham  Nah I don't think he goes to usf, he lives aro...
```

```
from sklearn.naive_bayes import MultinomialNB

# Initialize the model
model = MultinomialNB()
```

```python
# Train the model
model.fit(X_train_vectorized, y_train)

print("Model training completed.")

from sklearn.metrics import accuracy_score, precision_score, recall_score, f1_score,
confusion_matrix

# Predict on test data
y_pred = model.predict(X_test_vectorized)

# Evaluation Metrics
accuracy = accuracy_score(y_test, y_pred)
precision = precision_score(y_test, y_pred)
recall = recall_score(y_test, y_pred)
f1 = f1_score(y_test, y_pred)
conf_matrix = confusion_matrix(y_test, y_pred)

print("Model Evaluation:")
print(f"Accuracy: {accuracy:.4f}")
print(f"Precision: {precision:.4f}")
print(f"Recall: {recall:.4f}")
print(f"F1-Score: {f1:.4f}")
print("Confusion Matrix:")
print(conf_matrix)
```

```
Model Evaluation:
Accuracy: 0.9919
Precision: 1.0000
Recall: 0.9396
F1-Score: 0.9689
Confusion Matrix:
[[966   0]
 [  9 140]]
```

```python
def classify_email(email_text):
    """Classify an email as Spam or Not Spam"""
    email_vector = vectorizer.transform([email_text])
    prediction = model.predict(email_vector)[0]
    return 'Spam' if prediction == 1 else 'Not Spam'

def simulate_email_deletion(email_text):
    """Simulate spam deletion based on classification"""
    result = classify_email(email_text)
    if result == 'Spam':
        print(f"Deleted: {email_text}")
    else:
        print(f"Not Spam: {email_text}")

# Test Examples
sample_email_1 = "Congratulations! You've won a $1000 Walmart gift card. Click here to
claim now."
```

```
sample_email_2 = "Hi, can we schedule a meeting for tomorrow regarding the project?"


simulate_email_deletion(sample_email_1)
simulate_email_deletion(sample_email_2)
```

```
Deleted: Congratulations! You've won a $1000 Walmart gift card. Click here to claim now.
Not Spam: Hi, can we schedule a meeting for tomorrow regarding the project?
```

INFERENCE

The Naive Bayes classifier accurately classifies emails as spam or not spam based on text features. The model shows good performance in identifying spam emails and can effectively automate spam filtering. This system can be applied to real-time email filtering tasks.