Experiment No. 11

Aim: Case Study of Email Spam and Non-Spam Filtering using Machine Learning

Course Outcome: Analyze the mail filtering process

Theory:

Case Study: Email Spam and Non-Spam Filtering Using Machine Learning

1. Introduction

Email has become one of the most important communication tools, but it is often cluttered with unwanted messages called spam. Spam emails can carry advertisements, phishing links, or malware, affecting productivity and security. Machine Learning (ML) provides a robust solution to automatically classify emails as spam or non-spam (ham).

2. Objective

To design a system that can analyze incoming emails and accurately classify them into spam and non-spam, using machine learning techniques.

3. Dataset

The system uses a labeled dataset containing emails and their corresponding labels:

- **Spam:** Emails that are unsolicited or malicious.
- **Non-Spam (Ham):** Legitimate emails.

Example Datasets:

- Enron Email Dataset
- SpamAssassin Public Corpus

Each email typically contains:

- Subject
- Body content
- Sender information
- Metadata (date, headers)

4. Mail Filtering Process Using Machine Learning

The mail filtering process involves several steps:

Step 1: Data Collection

- Collect a large number of emails from various sources.
- Ensure the dataset is balanced with enough spam and non-spam examples.

Step 2: Preprocessing

- **Text cleaning:** Remove punctuation, HTML tags, special characters.
- **Tokenization:** Split email content into individual words or tokens.
- **Stop-word removal:** Eliminate common words (e.g., "and", "the") that do not contribute to classification.
- **Stemming/Lemmatization:** Reduce words to their root form (e.g., "running" → "run").
- **Feature extraction:** Convert text into numerical format using techniques like:
    - Bag of Words (BoW)
    - TF-IDF (Term Frequency–Inverse Document Frequency)

Step 3: Feature Selection

Identify key features that help distinguish spam from non-spam emails, such as:

- Frequency of suspicious keywords (e.g., “win”, “free”, “urgent”)
- Presence of hyperlinks or attachments
- Email header characteristics

Step 4: Model Selection

Common Machine Learning models for email classification include:

- **Naive Bayes:** Popular for text classification due to probabilistic approach.
- **Support Vector Machines (SVM):** Effective for high-dimensional text data.
- **Decision Trees / Random Forest:** Can handle categorical features like sender domain.
- **Deep Learning (LSTM / BERT):** Advanced models for semantic understanding of email content.

Step 5: Training

- Split the dataset into training and testing sets.
- Train the selected ML model on the training data to learn patterns of spam and non-spam emails.

Step 6: Evaluation

Evaluate the model using metrics such as:

- **Accuracy:** Overall correctness of classification.
- **Precision:** Fraction of emails classified as spam that are actually spam.
- **Recall (Sensitivity):** Fraction of actual spam emails correctly identified.
- **F1-Score:** Harmonic mean of precision and recall.

Step 7: Deployment

- Integrate the trained model into the email server or client system.
- Incoming emails are automatically filtered based on model predictions.
- Optionally, provide a feedback loop to continuously improve the model with user-reported misclassifications.

5. Advantages of Machine Learning in Email Filtering

1. Automatic detection without manual rules.
2. Adaptable: Learns new spam patterns over time.
3. Scalable: Can handle large volumes of emails efficiently.

6. Challenges / Limitations

1. Evolving spam techniques: Spammers constantly change tactics.
2. False positives: Legitimate emails incorrectly classified as spam.
3. Data imbalance: If spam is less represented in training data, model may underperform.
4. Resource-intensive: Advanced models like BERT require high computational power.

7. Example

Email Content:

Subject: *Congratulations! You won a prize*

Body: *Click this link to claim your $1000 reward now!*

Processing:

- **Preprocessing:** Remove punctuation, tokenize, stem.
- **Feature extraction:** Detect suspicious keywords (“won”, “prize”, “click”).
- **Model prediction:** Classifier labels it as spam.

Conclusion:

I have successfully done a case Study of Email spam and non-spam filtering using Machine Learning.