Spam\_Email Detection

Galal Osama : Sentiment Scores feature extraction

Supervisor: Dr. Wesam Ahmed

Group Number: 6

Submission date: May 12th, 2025

**Introduction**

Spam emails are unsolicited digital messages, often used for advertising, fraud, or malicious attacks. Detecting and filtering such messages is vital for both user protection and system integrity. This project presents a spam classification system using a **Logistic Regression model** trained solely on **sentiment-based features** extracted from email content. Instead of traditional word frequency techniques like TF-IDF, this approach explores the emotional tone and subjectivity of messages to distinguish between legitimate (ham) and spam emails.

**Chapter 2: Preprocessing**

Effective preprocessing ensures that the raw text is clean and structured for feature extraction. The following steps were applied:

* **Lowercasing**: Converts all characters to lowercase for consistency.
* **Punctuation Removal**: Removes symbols and special characters to reduce noise.
* **Stopword Removal**: Eliminates common non-informative words using NLTK’s stopword list.
* **Stemming**: Uses Porter Stemmer to reduce words to their root forms (e.g., "running" → "run").

The cleaned text is essential for accurate sentiment analysis in the next phase.

**Feature Extraction: Sentiment Scores Only**

**Sentiment Features**

* Extracted using the **TextBlob** library, which performs rule-based sentiment analysis.
* For each email message, two sentiment scores were computed:
  + **Polarity**: Indicates emotional orientation, ranging from -1 (very negative) to +1 (very positive).
  + **Subjectivity**: Measures opinion-based content, ranging from 0 (very objective) to 1 (very subjective).

These sentiment scores form a **2-dimensional feature vector** per message, used as input for the classifier.

**Model Training and Evaluation**

After converting the dataset into sentiment-based features, it was split into **training and testing sets** (80/20 split).

**Model Used**

* **Logistic Regression** classifier

**Evaluation Metrics**

The model’s performance was assessed using:

* **Accuracy**: Overall correct predictions
* **Precision**: Percentage of correctly identified spam messages out of all predicted spam
* **Recall**: Percentage of correctly identified spam out of all actual spam
* **F1-Score**: Harmonic mean of precision and recall
* **AUC Score**: Measures model’s ability to separate classes across decision thresholds

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Models | Training Accuracy | Testing Accuracy | Accuracy | Recall | Precision | F1-measure | AUC Value |
| Logistic Regression (FastText) | 99% | 97% | |  | | --- | |  |  |  | | --- | |  |   97% | 94% | 95% | 94.5% | |  | | --- | |  |  |  | | --- | | 98% | |
| Random Forest (FastText) | 99% | 96% | 96% | 92% | 93% | 92.5% | 97% |
| SVM | |  | | --- | |  |  |  | | --- | | 98% | | 95% | 95% | 91% | 92% | 91.5% | 97% |
| LSTM | |  | | --- | |  |  |  | | --- | | 96.5% | | 96% | 96% | 92% | 93% | 92% | 97% |

**Deep Learning**

An LSTM-based model was implemented using a FastText-initialized embedding layer.  
Architecture:  
- Embedding Layer (FastText, frozen)  
- LSTM (64 units)  
- Dropout (0.5)  
- Dense (1 neuron, sigmoid)