Spam\_Email Detection

Hassan Said: word2Vec Embedding 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 essential for both user protection and system integrity. This project presents a spam classification system using multiple machine learning models, including Logistic Regression, Random Forest, SVM, and a deep learning approach with LSTM, all trained on Word2Vec embeddings. Word2Vec represents words as dense vectors that capture semantic meaning, allowing the system to learn patterns indicative of spam.

**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 generating Word2Vec embeddings.

Feature Extraction: Word2Vec Embeddings

Instead of traditional frequency-based techniques (e.g., TF-IDF), this project uses Word2Vec embeddings to represent each email as a dense vector. Word2Vec is a two-layer neural network that processes text to learn word associations, creating a vector representation for each word in the vocabulary.

Word2Vec Embeddings

* Training Word2Vec Model: The Word2Vec model was trained using Gensim on the tokenized email content. It learns the associations between words, with each word being represented by a fixed-size vector.
* Embedding Generation: Each email’s vector is computed by averaging the embeddings of all words in the email, creating a fixed-size representation of the entire document.

These Word2Vec embeddings are then used as features for training various classification models.

Model Training and Evaluation

After converting the dataset into Word2Vec embeddings, it was split into training and testing sets (80/20 split).

Models Used:

1. Logistic Regression (Word2Vec embeddings)  
   Logistic Regression is used to perform binary classification (spam vs. ham) based on Word2Vec embeddings.
2. Random Forest (Word2Vec embeddings)  
   Random Forest, an ensemble method, is employed to improve classification accuracy by aggregating predictions from multiple decision trees.
3. SVM (Word2Vec embeddings)  
   Support Vector Machines (SVM) are used to find the optimal hyperplane that maximizes the margin between the two classes.
4. LSTM (Word2Vec embeddings)  
   A deep learning approach, Long Short-Term Memory (LSTM), is used to capture sequential dependencies in the text, which is important for text-based tasks like spam detection.

Evaluation Metrics

The model’s performance was assessed using:

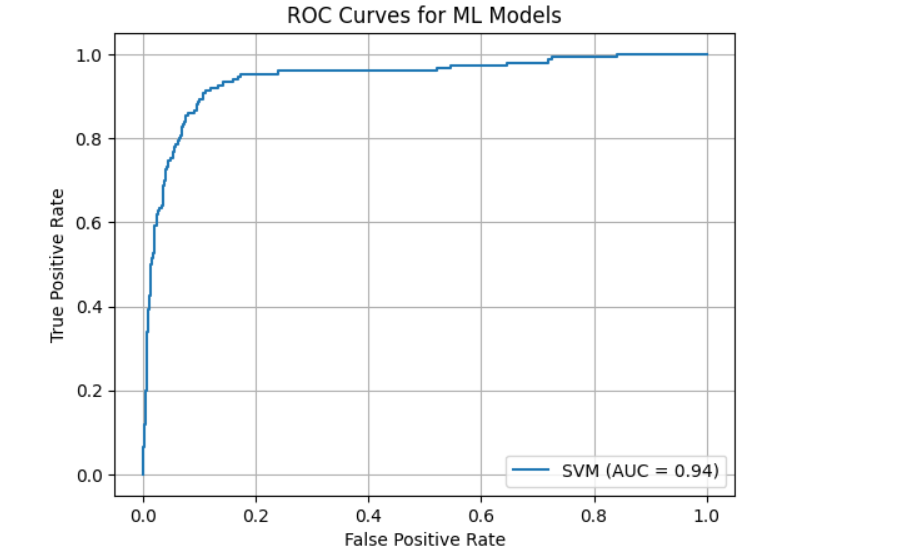
* Accuracy: Overall correct predictions.
* Precision: Percentage of correctly identified spam messages out of all predicted spam.
* Recall: Percentage of correctly identified spam out of all actual spam.
* F1-Score: Harmonic mean of precision and recall.
* AUC Score: Measures model’s ability to separate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

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Models | Training Accuracy | Testing Accuracy | Accuracy | Recall | Precision | F1-measure | AUC Value |
| Logistic Regression (FastText) | 99% | 97.5% | |  | | --- | |  |  |  | | --- | |  |   97.5% | 95% | 96% | 95% | |  | | --- | |  |  |  | | --- | | 99% | |
| Random Forest (FastText) | 99% | 97% | 97% | 93% | 95% | 94% | 98% |
| SVM | |  | | --- | |  |  |  | | --- | | 98% | | 96% | 96% | 91% | 94% | 93% | 98% |
| LSTM | |  | | --- | |  |  |  | | --- | | 98% | | 96% | 96% | 92% | 94% | 93% | 99% |

Visualization

A diagram of a logistic regression

AI-generated content may be incorrect.A diagram of a diagram

AI-generated content may be incorrect.

A diagram of a graph

AI-generated content may be incorrect.A blue squares with white text

AI-generated content may be incorrect.

**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)

Figure 5.1: LSTM ROC Curve