E-Mail Spam Classification

YZV 311E Term Project Proposal

Abdullah Bilici
Artificial Intelligence & Data Engineering
Istanbul Technical University
bilicia@itu.edu.tr
150200330

Bora Boyacıoğlu
Artificial Intelligence & Data Engineering
Istanbul Technical University
boyacioglu20@itu.edu.tr
150200310

***Abstract*—This documents is the project proposal for the YZV 311E Data Mining course term project. It is about an E-Mail classification model using techniques like Natural Language Processing. It is being held by Abdullah Bilici and Bora Boyacıoğlu, Istanbul Technical University. Team number is 8.**

***Index Terms*—email, spam, classification, natural language processing**

I. Introduction

In this project, we will generate a classification to classify spam E-Mails.

As the companies do not care about data security and sell people's data to other companies, frauds use this data to send spam E-Mails. These kinds of E-Mails sometimes look very realistic and most of the people think of it as a legitimate E-Mail. They may want people to enter their passwords, share their credit cards, and so on. However, most of the E-Mails are actually not very realistic. Though, sometimes people fall for their traps when they tell the people that they won money.

Therefore, classifying these E-Mails as spam is a critical task for E-Mail companies. There are some easy signs indicating that the E-Mail is spam, but there are also very professional spams. It is hard to separate them from legitimate ones.

One way to do this classification is actually to create a model using already human-classified data and look up the new E-Mails using this model. In the next section, dataset will be introduced.

Our team consists of two members: Abdullah Bilici and Bora Boyacıoğlu. We are both from Istanbul Technical University, Artificial Intelligence and Data Engineering department.

II. Dataset

We are going to be using a dataset from Kaggle. It is called **Email Spam Classification** (Harsh Sinha, 2020) [1] and contains basic E-Mail messages alongside a category *(spam/ham)*. Therefore, it only has two features: Text *(str)* and Spam *(bool)*. It contains 5728 values and has *8.95 MB* size. It uses *CSV* format.

All of the rows have a value **1** or **0** indicating whether the given E-Mail is spam or legitimate, and the message (content) of the E-Mail. Randomly selected six rows of the dataset are given in Table I.

TABLE I
E-Mail Classification Dataset

Text	**Spam**
subject : naturally irresistible your corporate identity lt is really hard to recollect a company : the market is full of suqgestions…	True
subject : greatly improve your stamina the longz system , both the capsules and the free instructional manual , give you the…	True
subject : new basis report bhavna : the basis report has been updated to cover 2000 prices . it is called basisnw 7 . xls and…	False
subject : localized software , all languages available . hello , we would like to offer localized software versions qerman , …	True
subject : approval for reviewer krishnarao , pinnamaneni v has suggested reviewers and submitted them for your approval . …	False
subject : research prc next steps the pep system is no longer available to accept feedback from reviewers . if necessary , …	False

III. Methodology

In this section, we outline a high-level methodology for building our email spam classifier. We'll provide an overview of the essential steps in our approach.

A. Text Preprocessing

To prepare the data for modeling, we will perform text preprocessing. This includes standardization tasks like removing special characters, converting text to lowercase, and tokenization. Additionally, we need to solve the issue of class imbalance.

- **Tokenization:** Splitting the text into words. We may use libraries like NLTK or spaCy for this.
- **Lowercasing:** Converting the text to only use lowercase characters for consistency.
- **Removing Special Characters:** Useing Regex to eliminate unnecessary characters (punctuations, tags).
- **Stopword Removal:** Eliminating words like and, the, in that seem extra.
- **Lemmatization or Stemming:** Reducing words to their base versions (*running* will be *run*).

B. Feature Engineering

In feature engineering, we will transform the preprocessed text data into a numerical format suitable for modeling. We will create feature vectors or representations that capture essential information from the emails.

- **TF-IDF (Term Frequency-Inverse Document Frequency):** Converting the text into numerical vectors using TF-IDF to show the importance of words in the texts.
- **Word Embeddings (e.g., Word2Vec, GloVe):** Creating dense vector representations of words and gather them to represent documents.
- **N-grams:** We may consider n-grams (sequences of n words) to capture context and phrases in the text.
- **Topic Modeling (e.g., LDA, NMF):** Extracting subjects from the emails and using their distributions as features.
- **Sentiment Analysis:** Analyzing the sentiment of the text as a feature.

C. *Data Splitting*

We will split our dataset into a training set and a testing set. This division is essential for assessing the performance of our classifier.

- **Train-Test Split:** Splitting the dataset into a training set (70% to 80% of the data) and a testing set (20% to 30%) to evaluate the model's performance.
- **Cross-Validation:** We may consider k-fold cross validation for more robust model evaluation.

D. *Model Selection and Training*

We will select and train a machine learning model suitable for text classification. The training process involves teaching the model to distinguish between spam and ham emails.

- Naive Bayes Classifier
- Support Vector Machines (SVM)
- Logistic Regression
- Random Forest

IV. Evaluation Methods

In our evaluation process, we will assess the success of our email spam classification project using the following steps:

A. *Performance Metrics*

We will employ standard evaluation metrics such as accuracy, precision, recall, F1-score, and AUC-ROC to measure the effectiveness of our models.

B. *Algorithm Selection*

We will experiment with different algorithms. We consider the following algorithms for email classification:

- Naive Bayes
- Logistic Regression
- Random Forest
- Support Vector Machines (SVM)

Our project's success will be judged based on the ability of the chosen algorithm to outperform the baseline model and achieve high performance metrics. The goal is to develop an effective and accurate email spam classification solution.

V. GitHub Repository

You can access our Github repository using this link: https://github.com/abdullahbilici/Group_8.

We have already sent invitations to teaching assistants. So, they will be able to see our progress. We also wanted to share the Personal Access Token for our repository: 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

We are going to be updating the file `Progress.txt` in order to show our progress.

VI. Time Plan & Distribution of Work

We separated and listed our tasks in a timeline in Table II for better visuality. You can find the explanations of each task below the chart.

TABLE II
Timeline and Work Distribution

	Week									
	1	2	3	4	5	6	7	8	9	10
Abdullah	■	■	■	■	■	■	■	■	■	■
Text Preprocessing	■	■								
Feature Modelling			■	■	■					
Model Selection						■				
Model Training							■	■	■	
Conclusion									■	■
Bora	■	■	■	■	■	■	■	■	■	■
Text Preprocessing	■	■	■							
Context Analysing				■	■					
Data Splitting						■				
Model Training							■	■	■	
Reporting									■	■
Common	■	■	■	■	■	■	■	■	■	■
Analysis & Featuring	■	■	■	■	■					
Splitting & Modelling						■	■	■	■	■

Abdullah Bilici

1) Text Preprocessing *(Lowercasing, Stopword Removal)*
2) Feature Modelling *(Transforming the text into numerical format and vectorising them)*
3) Model Selection *(Selecting an ML model suitable for text classification)*
4) Model Training *(Training the model)*
5) Conclusion *(Briefly conclusing what we have accomplished)*

Bora Boyacıoğlu

1) Text Preprocessing *(Tokenization, Removing Special Characters, Lemmatization or Stemming)*

2) Context Analysing *(Capturing the context and phrases and analysing them to use as features)*
3) Data Splitting *(Train-Test Split)*
4) Model Training *(Training the model)*
5) Reporting *(Reporting our results and conclusions)*

Common Tasks

1) Data Analysis and Feature Engineering *(Understanding our data to work on it and generating its features)*
2) Data Splitting and Modelling *(Understanding our data to work on it and generating its features)*

VII. Conclusion

Our project uses data mining techniques to address the serious issue of spam E-Mails. It aims to improve E-Mail security by developing a classification model that can distinguish between legitimate and spam emails, as spam emails are getting more complex and posing serious threats to consumers. We will use various techniques, including text preprocessing and feature engineering, evaluate our model using multiple metrics, and experiment with different algorithms. A GitHub repository has been set up for transparency and collaboration. Thank you for your helps.

References

[1] Harsh Sinha, Email Spam Classification, 2020, Kaggle https://www.kaggle.com/code/harshsinha1234/email-spam-classification-nlp/input