



Enhancing Email Management through Machine Learning: A Multi-Label and Priority Approach

Prepared By

Name: Abubaker Hussein

ID No.: 253002549

Submitted in fulfillment of the requirements for the degree Master of
Science in Artificial Intelligence

At

College of Informatics

Midocean University

UAE

Supervisor Name: Dr. Amjad Qatish

College of Informatics, Midocean University, Comoros

Email: amjadqtaish@midocean.edu.km

Acknowledgements

In the first place, I am thankful to Almighty Allah for giving me the strength, patience, and perseverance to conduct this research.

I want to thank my supervisor, Dr. Amjad Qatish, for his continuous guidance, valuable suggestions, and support throughout all the stages of this research. The support and guidance of Dr. Qatish are of paramount importance for the successful completion of this thesis.

In addition, I would like to thank the faculty members of the College of Informatics at Midocean University for providing an inspiring environment and knowledge that helped me formulate this thesis.

Finally, I would like to thank my family for their love, patience, understanding, and encouragement. Without their encouragement and motivation, it would have been impossible for me to reach this milestone.

This research could not have become a reality without the support and encouragement of all those people who helped in the formulation of this research directly or indirectly.

Abstract

Email management is becoming increasingly difficult due to the growing volume and complexity of organizational email communications. Current email categorization methods mainly focus on individual tasks, such as spam filtering or single-label categorization, limiting their effectiveness in handling emails that contain multiple topics and different priority levels.

This study presents a multi-task learning model based on a DistilBERT encoder for simultaneous multi-label email classification and priority prediction. The proposed architecture employs a shared DistilBERT encoder followed by two task-specific output heads.

The model was trained on a cleaned and labeled subset of the Enron Email Dataset using a stratified Train–Validation–Holdout split. Experimental results showed that the proposed DistilBERT model achieved a Macro F1-score of 0.9672 and a Hamming Loss of 0.0175 for category classification, and an Accuracy of 95.29% with a Macro F1-score of 0.9327 for priority prediction.

In addition, threshold optimization improved multi-label classification performance, while LIME enhanced model interpretability by identifying the most influential features contributing to each prediction.

List of Publications

1. Hussein, A. *A Multi-Task DistilBERT Framework for Multi-Label Email Classification and Priority Prediction*. Under review.

Table of Contents

Acknowledgements.....	iii
Abstract.....	iv
List of Publications	v
Table of Contents.....	vi
List of Abbreviations	viii
Chapter 1 Introduction	9
Overview 1.1	9
Problem Definition 1.2.....	9
Thesis Objectives 1.3.....	10
Thesis Contribution 1.4	10
Ethical and Privacy Considerations 1.5.....	11
Thesis Organization 1.6.....	11
Chapter 2 Background and Related Work.....	12
Introduction 2.1.....	12
Related Work 2.2	12
Comparative Analysis 2.3.....	18
Research Gap 2.4	22
Summary 2.5.....	23
Chapter 3: Research Methodology.....	24
Introduction to the Chapter 3.1	24
Research Methodology 3.2	25
Dataset Description 3.3.....	25
Data Preprocessing 3.4	27
Data Labeling Strategy 3.5.....	30
Dataset Splitting 3.6	36
Baseline Models 3.7.....	38
Chapter 4: Implementation and Results	43
Introduction to the Chapter 4.1	43
Experimental Environment 4.2.....	43
Data Preparation for Implementation 4.3	44

Implementation of Baseline Models 4.4	45
Implementation of the Proposed DistilBERT Model 4.5	50
Model Evaluation Using the Holdout Test Set 4.6	55
Threshold Tuning 4.7	59
Explainable Artificial Intelligence (XAI) 4.8	62
Final Model Comparison 4.9	65
Discussion 4.10	72
Chapter Summary 4.11	75
Chapter 5: Conclusions, Recommendations, and Future Work	76
Introduction 5.1	76
Conclusions 5.2	76
Study Contributions 5.3	77
Recommendations 5.4	78
Future Work 5.5	79
Chapter Summary 5.6	80
References	81

List of Abbreviations

- **Acc.:** Accuracy
- **ANN:** Artificial Neural Network
- **ANOVA:** Analysis of Variance
- **ROC:** Receiver Operating Characteristic
- **NLP:** Natural Language Processing
- **TF-IDF:** Term Frequency–Inverse Document Frequency
- **SVM:** Support Vector Machine
- **LR:** Logistic Regression
- **XAI:** Explainable Artificial Intelligence
- **LIME:** Local Interpretable Model-agnostic Explanations
- **MTL:** Multi-task Learning

Chapter 1 Introduction

1.1 Overview

In today's modern digital world, emails represent the predominant method of communication in all aspects of life, ranging from business to educational institutions and even customer support services (Iqbal & Khan, 2022). Due to the huge number of emails received daily, individuals and firms suffer a lot in dealing with them through the processes of management, classification, and response (Bhalla et al., 2021). The issue is no longer just about removing spam messages; however, it now requires much more efficient means of sorting and classifying the emails based on their genre and urgency (Hirpara & Polara, 2025).

Since classical methods of email management require manual effort or simple rule-based algorithms, modern technologies require more advanced approaches in this aspect (Khare et al., 2022). To overcome this problem, machine learning and natural language processing techniques have been used to enhance classification efficiency and boost productivity. In recent years, some advances in deep learning and transformer models have been made to deal with academic email management (Duong et al., 2025) and even multi-label classification in industrial or financial spheres (Jain et al., 2024).

1.2 Problem Definition

Although modern spam detection filters demonstrate high accuracy, they generally consider emails either as binary (Ham/Spam) or belonging to one fixed category only (Bhalla et al., 2021). Professional emails have several facets at once because there might be an invoice, a problem related to work, and an urgent request in the same letter (Jain et al., 2024). Thus, Multi-Label Classification (assigning multiple labels to a text), together with Automatic Priority Assessment (determining the importance of a message), which is not addressed currently by literature on the issue (Wang et al., 2024; Chuttur & Parianen, 2022), should become an integral part of any effective automated system. This will allow handling multiple aspects of a task and prioritizing them accordingly to ensure prompt response to the important emails.

1.3 Thesis Objectives

The main aim of this thesis will be the development of a machine learning approach that can improve email management through the consideration of the complexities and urgency involved. The specific research aims will involve:

1. An evaluation of various machine learning algorithms utilized in email categorization and prioritization
2. The design of a Multi-Label Classification system that can identify different intents from one email message.
3. The development of a priority ranking algorithm that will prioritize emails depending on their classified intents and urgency.

1.4 Thesis Contribution

The key developments made by this research towards automating email management are categorized as follows:

- **Unified Architecture for Multi-Intent & Urgency Detection:** The primary theoretical contribution is the design of a unified framework that synchronizes multi-intent detection with urgency ranking. This architecture addresses the gap in existing literature where these two critical dimensions of email management were previously treated as isolated tasks.
- **Contextual Superiority Over Traditional Filtering:** The key empirical contribution lies in overcoming the limitations of linguistic diversity. By leveraging deep contextual embeddings, the proposed model demonstrates superior performance over rigid, keyword-based filters in understanding complex and multifaceted professional communications.
- **Methodological Efficiency and Explainability:** Through the incorporation of Explainable AI (XAI) techniques, the research enhances the transparency of automated decision-making. This approach ensures that priority assignments are interpretable while maintaining high performance even in data-constrained environments with minimal labeled datasets.
- **Functional Proof-of-Concept Prototype:** On a practical level, the thesis delivers a functional prototype that validates the real-world applicability of the proposed models, proving their effectiveness in streamlining email workflows within professional and industrial settings.

1.5 Ethical and Privacy Considerations

Because email exchanges can involve private matters, this study follows careful ethical practices. The material drawn from the Enron collection is already public and stripped of identifying details. During preparation steps, names or unique markers do not appear, while every test aligns with rules meant to protect business message confidentiality. Though sourced from real workplaces, the information stays shielded throughout each phase. From start to finish, respect for privacy shapes how work proceeds. Not a single person's identity surfaces once processing begins. Rules guiding organizational data use remain central throughout. Even though methods change slightly across trials, protection never weakens. What emerges reflects patterns, not individuals. Each stage avoids exposure by design. Confidentiality holds firm at all points. The outcome does not represent any individual but rather reflects the patterns of the communication process. Every step in the process ensures anonymity and confidentiality.

1.6 Thesis Organization

The remainder of this thesis is structured as follows:

- **Chapter 2** provides the theoretical background and reviews recent studies on email classification, priority prediction, and explainability.
- **Chapter 3** presents the proposed methodology, including dataset preparation, model architecture, and evaluation metrics.
- **Chapter 4** reports the experimental results and discusses the performance of the proposed system.
- **Chapter 5** concludes the thesis and outlines directions for future research.

Chapter 2 Background and Related Work

2.1 Introduction

This chapter covers background and related work that is relevant to understanding the development of an email classification system. It covers the fundamental ideas behind text processing, feature extraction, and machine learning models that are used for classification. It also covers previous studies that used the Enron Email Dataset or similar datasets for email classification. Finally, this chapter covers the gap that this research seeks to fill.

2.2 Related Work

In their study, Fareed et al. (2025) test 19 machine learning classifiers in terms of spam detection utilizing TF-IDF features from the email spam dataset, which contains 7,673 emails. With pre-processing and a 70/30 train-test split performed on the data, the models were evaluated according to accuracy, precision, recall, F1-score, and variance-based stability measures, stressing generalization over performance.

Even though some of the models scored highly in terms of accuracy, the Random Forest and logistic regression models turned out to be the most robust ones in terms of variance, achieving 0.9818 and 0.9845 accuracy, respectively. The Decision Tree model was particularly susceptible to overfitting in comparison with other models. Nevertheless, the paper offers an interesting perspective into building a robust classifier; however, it fails to discuss issues concerning multi-labeling and the prioritizing of emails—important problems for advanced spam filters.

Hirpara and Polara (2025), in their research paper, outline an AI-driven framework to optimize the handling of company emails through automation of classification, prioritization, and response generation. The emails are extracted through IMAP/SMTP protocols and then pre-processed using libraries such as NLTK and spaCy, or cleaning, tokenization, lemmatization, and TF-IDF vectorization.

In particular, the framework uses a Decision Trees classifier for categorization, VADER/TextBlob and rule-based techniques for urgency detection, and Gemma 1.1 2B instruction-tuned model for generating responses automatically.

When tested using a company email dataset at Elecon, the framework yielded impressive results, including an accuracy of 0.93 for classification, 0.73 for urgency detection, and an acceptability rate of 83% for the responses generated. However, when it comes to managing email with a multi-label and priority-focused approach,

InboxG is not adequate due to its single-label category predictions and simplistic priority determination; hence, the need for a multi-label architecture and more comprehensive priority inference.

Thu Duong et al. (2025) describe AEEM, which is a deep learning framework used for classifying emails sent by students in higher education institutions. This helps prevent email overload and enhances the efficiency of administration.

Based on the five categories in which Vietnamese universities categorize emails, the proposed framework uses preprocessing techniques, keyword extraction using the RAKE method, text summarization using the ViT5 model, and classification using a PhoBERT-based Transformer model, optimized via AdamW.

The five-fold cross-validation method is utilized for evaluation purposes. The model exhibits 93% overall accuracy, with the PBKS and PBS models being equally accurate. Despite its efficiency, the model is limited in terms of its single-label nature, computational intensity to be deployed in real time, and limitations due to the dataset used, especially for the less represented labels.

U. Jain et al. (2024) introduced a model architecture for multi-label and multi-class text classification, focusing on improving named entity recognition (NER) in multi-class documents like financial emails, which contain several classes (such as invoices and payments). In the research, two large-scale data sets were used, including 800,000 financial emails with five classes and 500,000 customer complaints with ten classes.

The architecture consisted of pre-training a transformer using datasets with labeled classes according to the IOB (Inside, Outside, Beginning) annotation method for NER tasks and then fine-tuning using a better encoder-decoder attention algorithm to improve the mapping of classes in a multi-label document.

After that, knowledge distillation was performed to simplify the model and speed up the prediction process.

The model showed great results, providing 94% accuracy for perfect multi-label predictions and 97% accuracy for each class identification. However, despite the improvements, the model lacks a priority-based label selection policy and ignores the effect of knowledge distillation on multi-label relationships.

In their work, Qunbo Wang et al. (2024) introduce BEAL as a deep active learning method designed to boost multi-label text classification tasks without relying heavily on large-scale annotated datasets. By using Bayesian deep learning techniques, BEAL calculates an estimate of the posterior predictive distribution via the

application of the expected-confidence acquisition function, which chooses the most informative and uncertain examples to label.

For the evaluation purposes of BEAL, the authors used the AAPD and StackOverflow datasets and employed a fine-tuned BERT-based MLTC model. According to the results achieved, BEAL performed better than any other active learning algorithm and delivered results comparable to those of full-scale training by using merely 64% of labeled samples on the AAPD dataset and 40% on StackOverflow.

In addition, the authors attribute the effectiveness of their method to the precision of the uncertainty estimation that leads to the improvement of the sample selection process. Despite the accuracy gained, BEAL suffers from computational costs since it involves several forward passes and does not seem to be useful when applied to tasks related to priority inference in domain-specific environments, such as emails.

Adnan et al. (2023) discuss the development of a stacked learning model that seeks to optimize the efficiency of spam detection while overcoming the drawbacks of existing single-classifier models. The Spam-Assassin and Enron-Spam databases were combined by the researchers, resulting in a collection of 13,629 emails. Preprocessing was applied to the data, which included normalization, tokenization, removal of stop words, and vectorization using the TF-IDF approach.

Five classifiers, namely logistic regression, decision tree, k-nearest neighbors, Gaussian Naive Bayes, and AdaBoost, were trained, and their predictions were combined by means of a logistic regression classifier using grid search and cross-validation. The stacking technique showed higher performance and obtained 98.8% accuracy, 98.8% recall, and an F1-score of 98.9%, outperforming any other individual classifier.

In view of the current research on multi-label classification and email prioritization, the discussed work can be considered as being constrained by the focus on binary spam detection only.

EmFore is an Internet-based tool developed by Singh et al. (2023) that employs symbolic and predicate-based rules to classify emails based on the user's movement of emails to specific folders.

This tool is meant to overcome the limitations of machine learning techniques, which are hard to understand and inefficient in inductive logic programming, which takes time to learn.

It is characterized by rule refinement through specialization and generalization, ranking of predicates through folder abstractions and similarities, and the suppression of rules to minimize false positives.

Tested on the Enron and Avocado datasets, which consist of more than 130,000 emails that are arranged into several thousand folders, it has proven superior to other techniques, including neural and ILP-based ones.

While EmFore has demonstrated its effectiveness compared to 13 other models, it is limited by its binary nature and lack of support for prioritization of emails, which is what this thesis aims to develop.

Khare et al. (2023) introduce a solution that automates email management using Robotic Process Automation (RPA) technology and artificial intelligence by performing operations such as user login, filtering valuable emails, categorizing emails based on pre-specified labels, downloading attachments, and deleting unnecessary emails. In the proposed approach, incoming emails are processed by the native email server and Email Processor Engine, where email metadata are recorded, contents are extracted by web scraping, and data enrichment techniques are used to filter out noise. This system was developed in UiPath, where the assistant mainly uses keywords to classify emails and employs the RPA algorithm to perform other tasks.

The experimental results show that the bot performs much faster than manual efforts, with nearly ten times the speed of human intervention and achieving nearly 100% accuracy. Nonetheless, from the perspective of multi-label and priority-based email management, the model is limited by the rule-based and single-label design of the assistant.

The study by Chuttur et al. (2023) explored the use of machine learning approaches in the classification of emails based on emotions. As a part of tackling organizational problems associated with the overload of emails and their consequences on employee productivity and customer experience, the authors applied the NRC Emotion Lexicon in combination with the preprocessing techniques applied to over 500,000 emails from the Enron data set.

Based on the emotion intensity, the authors assigned emails to high-priority, low-priority, or neutral categories and assessed four supervised learning models (Multinomial Naive Bayes, Support Vector Machine, Logistic Regression, and Random Forest). The best results were shown by Logistic Regression and Linear Support Vector Classification, which scored 72% of accuracy, while Multinomial Naive Bayes was the least accurate, and Random Forest demonstrated overfitting. In

terms of applicability, emotion-based classification does not meet real-world demands due to low efficiency and focus on English-language material and three classes only. In the case of multi-label and prioritized emails, more advanced techniques are necessary.

Francisco Jáñez-Martino et al. (2022) in their present a comprehensive study analyzing the constantly changing tactics used by spammers and dataset shift issues in spam detection systems. Using seven different data sets, which were collected between 2000 and 2018, the researchers examine four types of spam filters that combine TF-IDF or Bag-of-Words encoding methods with Naïve Bayes or SVM classifiers.

While models tested on the same dataset have an accuracy level above 95%, their performance drastically degrades under the conditions of dataset shift, resulting in error rates of 48.81% for filters designed based on older datasets applied to more recent spam. These findings confirm the vulnerability of spam filters to the changes in spammer tactics; however, the study is limited by its exclusive focus on binary spam classification and does not address the broader challenges of categorizing diverse email content or inferring priority key components of the current thesis.

Kaddoura et al. (2022) surveyed spam detection techniques across different platforms such as emails, Facebook, YouTube, Twitter, and review websites. Some data collection techniques that are commonly used, according to Kaddoura et al. (2022), include those based on the usage of APIs and publicly available data sources such as Spam-Assassin, Enron-Spam, BuzzFeed News, and TripAdvisor, while some of the most typical pre-processing techniques include tokenization, normalization, stemming, lemmatization, and removal of stopwords.

Spam classification techniques discussed by Kaddoura et al. (2022) can be divided into rule-based systems, machine learning models (for example, SVM, Naïve Bayes, decision trees, K-NN, and random forest), as well as deep learning systems (including CNNs and LSTMs). It should also be noted that some deep learning algorithms reach high accuracies of even up to 98.39%, but despite revealing some valuable information, the authors mention certain problems associated with changes in spam, low-quality datasets, and high computational complexity.

In regard to multi-label and priority-based classification, the present research also focuses on binary and single-label spam classification without discussing priority inference or multi-label classification techniques.

Iqbal and Khan (2022) explore the issue of spam identification based on advanced machine learning algorithms and efficient feature selection techniques to underline

the significance of selecting the right attributes to differentiate spam from ham. With respect to 4,601 instances having 57 features included in the UCI Spam-base database, normalization was applied, and Point-Biserial correlation was used for feature selection, which was identified as an innovative approach used in such applications.

Several classifiers, including KNN, SVM, Random Forest, Decision Tree, ANN, Logistic Regression, and RBF, were tested with the performance measured through 10-fold cross-validation. Among the examined approaches, ANN and SVM yielded the best results with accuracies of 98.06% and 98.00%, respectively. Still, considering the multi-label nature and priority-based classification, this study is restricted because it relies only on the binary classification of emails.

Ahmed et al. (2022) give an exhaustive discussion on supervised learning, unsupervised learning, and reinforcement learning approaches applied for spam filtering in emails and IoT systems. The article highlights the common data preprocessing procedures and different kinds of models used in spam detection, including the Naïve Bayes algorithm, Support Vector Machines (SVM), Decision Trees, K Nearest Neighbor (KNN), Neural Networks, Extreme Learning Machines, and clustering algorithms.

These models have achieved high accuracy levels of more than 95%, with some SVM and Naïve Bayes models attaining accuracies exceeding 99% scenarios. However, despite the numerous advancements, the study reveals various drawbacks in this area, which include the use of artificial datasets, the inability to adapt to new spam types, high false-positive rates, and the failure to incorporate multiple languages and image spam detection. With regards to multi-label and prioritization in email management, this paper discusses only spam filtering in binary categories without considering multi-category spam and prioritization.

Said Salloum et al. (2022), in their paper entitled "Systematic Literature Review on Phishing Email Detection," analyze the results from 100 articles published from 2006 to 2022 on NLP approaches for phishing email detection. This review discusses common preprocessing and feature engineering practices like TF-IDF and word embeddings, with the Nazario corpus found as the most popular corpus for the purpose.

The classification algorithms commonly employed include SVM, Naïve Bayes, Decision Trees, Random Forest, and Deep Learning approaches such as Recurrent Neural Network (RNN) and Convolutional Neural Networks (CNN), measured based on accuracy, precision, recall, and F1-score metrics. Most of the literature employs

Python-based approaches and considers email content analysis with email headers and URLs. Nonetheless, the review has indicated that there is a significant lack of literature related to Arabic phishing emails and an excessive focus on binary classification. Regarding multi-label classification and priority-based email management, the review is silent on multi-classification and priority inference approaches, which the present thesis aims to solve.

Ali et al. (2021) propose a method for categorizing emails into order, request, and general classes by using the approach to imperative sentences as a way to solve email overload. Based on a sample of 900 emails, preprocessed by tokenization, stop word removal, and Word2Vec embedding, the authors trained CNN and RNN classifiers. The findings reveal that RNN demonstrates higher performance (94.9%), compared to CNN and Fuzzy ANN, and that its accuracy improves when the amount of training samples grows. Although successful for emails classified based on imperatives, this methodology is confined by its restrictive three-class framework, dependency on explicit imperatives, and relatively small database.

In terms of multi-label and priority-based email sorting, which are the main tasks tackled in the proposed work, the method does not account for emails belonging to multiple categories or requiring urgency-based prioritization, key challenges addressed in the current thesis.

In Vinod Kumar Bhalla et al. (2021), there is a discussion about machine learning algorithms used for genre classification of emails to enhance their organization and facilitate more efficient search. 5,000 emails obtained from the Enron collection were preprocessed through the removal of stop words and stemming before being assigned weights based on their keywords using the TF-IDF method to train a Support Vector Machine (SVM) classifier with the one-versus-rest technique, which enabled the use of only positive training samples.

The proposed solution was very successful, achieving 97% accuracy when predicting official genres, 87% for personal, 93% for promotional, and 92% for confidential emails. From the perspective of multi-label and priority-based email management, the study does not address scenarios where emails span multiple genres or require urgency-based ranking, key challenges that the current thesis seeks to overcome.

2.3 Comparative Analysis

The following table summarizes the core methodology, dataset, contributions, and limitations of the reviewed studies:

Table (2.1): Comparative Analysis of Literature Review Studies

Author (Year)	Paper title	Description	Dataset	Methodology / Technique	Results	Advantages	Disadvantages
Adnan et al. (2023)	Improving spam email classification accuracy using ensemble techniques	Using Stacking approach to improve spam detection.	Aggregated Spam Datasets	Stacking Ensemble (LR, DT, KNN, NB, AdaBoost)	Achieved a peak accuracy of 98.8%.	Reduces overfitting by merging multiple models.	Higher resource consumption due to multiple models.
Ahmed et al. (2022)	ML Techniques for Spam Detection in Email and IoT Platforms	Challenges of spam in Email and Internet of Things (IoT).	Analytical Review	ML Algorithms Comparison	Identified research gaps in securing IoT devices from spam.	Links Email security with the modern IoT ecosystem.	Very broad scope; lacks depth in Email-specific solutions.
Ali et al. (2021)	Feature Extraction aligned Email Classification based on Imperative Sentences	Classification based on identifying "imperative" (command) sentences.	Enron Dataset	DL (CNN/RNN) + Imperative Sentence Selection	Improved understanding of sender intent and functional emails.	Innovative focus on "verbs" and actionable content.	May misclassify informal or purely social emails.
Bhalla et al. (2021)	Automatic Genre Categorization of Emails into predefined categories	Classifying emails by genre (Official, Personal, Business).	Enron and others	TF-IDF + SVM	Accuracy exceeds 90%.	Simplicity and high execution speed.	Rely heavily on predefined keyword lists.
Chuttur et al. (2022)	A Comparison of Machine Learning Models to Prioritize Emails using Emotion Analysis	Prioritizing emails based on customer emotion analysis.	Customer Service datasets	Emotion Analysis + (SVM, Naive Bayes, Random Forest)	Random Forest achieved the best accuracy at 84%.	Integrates emotional context into priority queuing.	Limited scope (focused only on customer service).
Duong et al. (2025)	AEEM: A Deep Learning Approach to Automated Educational Email Management	System for managing academic emails for students.	University-specific data (Vietnam)	Deep Learning + Text Summarization	93% accuracy with significant improvement in class balance.	Combines summarization with classification for better UI.	Localized dataset; might not generalize to other languages.
Fareed et al. (2025)	ML-Based Email Spam Detection: Accuracy,	Analyzing model robustness and	Public Ham/Spam Dataset	19 Classifiers + Robustness Analysis	Random Forest was found to be the	Extensive comparison across 19 different algorithms.	Focuses mostly on classic ML rather than Neuro-AI.

	Overfitting, and Robustness	resistance to overfitting.			most stable and robust.		
Hirpara & Polara (2025)	InboxG – Smart Mail Classifier	An AI system for manufacturing companies to detect urgency.	Elecon Engineering Co. Data	Gemma 1.1 2B (LLM) + Decision Tree	Accurate urgency detection and intelligent autoreplies.	Utilizes modern Large Language Models (LLMs).	Requires high hardware specs for LLM inference.
Iqbal & Khan (2022)	Email classification analysis using machine learning techniques	Feature selection analysis to improve spam detection.	UCI Spambase	Point-Biserial Correlation + SVM & ANN	98.06% accuracy using Artificial Neural Networks (ANN).	Feature selection methodology.	Uses a relatively old dataset (UCI Spambase).
Jáñez-Martino et al. (2022)	A review of spam email detection: analysis of spammer strategies	Comprehensive review of spammer strategies and the "Dataset Shift" problem.	SLR (Systematic Literature Review)	SLR Methodology	Identified "Dataset Shift" as the main challenge for filter robustness.	In-depth analysis of adversarial behavior (spammers).	Theoretical study lacks new experimental benchmarking.
Kaddoura et al. (2022)	A systematic literature review on spam content detection and classification	Review of spam in social media (FB, Twitter, YT) and Email.	Multi-platform review	SLR	Email remains the most targeted medium for spam content.	Compares spam behavior across different platforms.	Broad focus reduces specific technical depth for email.
Khare et al. (2022)	E-Mail Assistant – Automation using Robotic Process Automation (RPA)	Automating email handling (attachments, sorting, replies).	Experimental setup	RPA (UiPath) + AI	Successfully reduced human effort in routine email tasks.	Practical integration with corporate office environments.	Dependent on commercial tools (UiPath).
Salloum et al. (2022)	A Systematic Literature Review on Phishing Email Detection Using NLP	Systematic review of NLP techniques used in detecting phishing emails.	SLR (2006-2021 period)	SLR + NLP Techniques Analysis	DL models are starting to outperform traditional ML in URL/link detection.	High comprehensiveness across a long timeframe.	Does not cover the latest Transformer-based models (2023+).
Singh et al. (2023)	EmFore: Online Learning of Email Folder Classification Rules	Continuously learning folder classification rules.	Microsoft Internal Data	Online Learning + Rule-based	Continuous improvement in folder filing accuracy over time.	Adapts dynamically to changing user behavior.	Requires a constant stream of data to remain effective.

Wang et al. (2024)	Deep active learning for multi-label text classification	Deep active learning for multi-label classification.	Reuters, AAPD (Benchmark)	Bayesian DL + Active Learning (BEAL)	Reduced labeled data requirements by 30% with stable accuracy.	Significantly lowers manual labeling costs.	High computational complexity of Bayesian models.
--------------------	--	--	---------------------------	--------------------------------------	--	---	---

2.4 Research Gap

Current research reveals a noticeable gap in how email management is handled, with approaches often scattered and incomplete. Reviews from 2022 (Jáñez-Martino et al.; Salloum et al.; Kaddoura et al.; Ahmed et al.) mainly focus on problems like dataset shifts and detecting threats like Spam and Phishing through simple binary classification. They often ignore how to organize and prioritize everyday, legitimate emails.

Some studies (Bhalla et al., 2021; Iqbal & Khan, 2022; Fareed et al., 2025) have developed accurate models to sort emails by topic and reduce overfitting, but they only look at single-label classification.

That overlooks the fact that many emails cover several topics at once; for example, one email could be about both an invoice and a technical problem. Ali et al. (2021) addressed this somewhat by extracting important sentences, but they didn't provide a full system to handle multi-intent emails.

Other work focuses on improving workflow. Khare et al. (2022) use Robotic Process Automation, Singh et al. (2023) try to learn folder rules on the fly, and Duong et al. (2025) look at managing educational emails. Still, they mostly leave out how to adjust priority levels dynamically.

A few researchers (Chuttur & Parianen, 2022; Hirpara & Polara, 2025) have looked at priority by analyzing emotions or urgency with Large Language Models, but they treat priority as a separate feature instead of connecting it to emails that have multiple intents.

The biggest gap is the lack of a system that combines multi-label classification and automatic priority setting in a consistent way. Even recent advances (Wang et al., 2024; Jain et al., 2024) focus on sophisticated multi-label methods using Transformers and Active Learning but haven't applied them to smart priority management.

This thesis seeks to fill that gap by building a model that uses multiple email labels and automatic priority assignments, aiming to provide a better tool for managing professional workflows.

2.5 Summary

This chapter reviewed the theoretical foundations and recent advancements in email classification, priority prediction, and explainability. The analysis highlighted the strengths and limitations of existing approaches and identified the need for a unified, transparent framework, setting the stage for the methodology presented in Chapter 3.

Chapter 3: Research Methodology

3.1 Introduction to the Chapter

The following chapter gives an overview of the methodology used in this research for developing and evaluating the proposed email intelligence system. The description of the research process starts with the stage of dataset preparation and pre-processing, automatic labeling method, data splitting, baselines' implementation, DistilBERT multi-task model development, experiment configurations, and evaluation criteria.

The methodology used in this research was constructed in a way to provide similar experimental conditions for training and evaluating all models, which would allow a comparison of traditional machine learning methods and the proposed deep learning approach.

Figure 3.1 Research Methodology Workflow

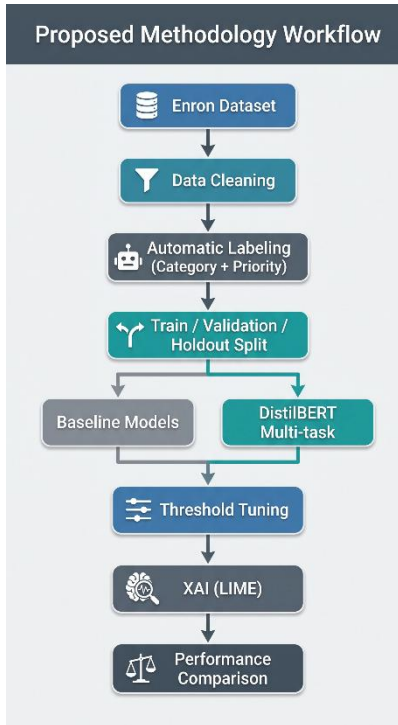


Figure 3.1 shows the methodology of research followed in this thesis, starting with dataset preparation and finishing with evaluation and comparison of models.

3.2 Research Methodology

Figure 3.1 shows the general procedure that was used in this research. The method involves collecting and preparing the email dataset, after which the automatic data labeling process categorizes the emails. In addition, data labeling is done on the priority prediction. The prepared dataset is divided into training, validation, and hold-out sets.

First, two machine learning baselines are developed to set a level of performance. This is followed by training the proposed DistilBERT multi-task model on the dataset. Threshold tuning is done after training to optimize multi-label category prediction, followed by Explainable Artificial Intelligence (LIME) to explain predictions. All the models are compared based on the same performance metrics. This structure ensures that the experiment proceeds progressively from one step to another.

Table (3.1) Research Methodology Stages

Stage	Description
Dataset Preparation	Load and preprocess the Enron email dataset
Data Cleaning	Remove duplicates and invalid emails
Automatic Labeling	Generate category and priority labels
Dataset Splitting	Create Train, Validation, and Holdout sets
Baseline Models	Logistic Regression and Linear SVM
Proposed Model	DistilBERT Multi-task Learning
Threshold Tuning	Optimize category decision threshold
Explainability	LIME-based prediction interpretation
Final Comparison	Compare all experimental results

The various stages involved in the research methodology adopted in this study are summarized in Table 3.1. Each of the stages plays an important role in developing and evaluating the proposed email classification model.

3.3 Dataset Description

The performance of any machine learning model relies heavily on the quality of the dataset utilized during training and testing. Hence, choosing a suitable dataset was one of the necessary steps that had to be considered in this research to enable the proposed model to acquire relevant features about email categorization and prioritization.

This research involved the use of the Enron Email Dataset, which is one of the most popular datasets employed in studies that involve email classification and natural language processing. This dataset comprises a huge number of genuine emails that were exchanged between employees working in the Enron Corporation before its collapse. As these emails constitute real business messages, they cover various topics including administrative queries, legal matters, finances, meetings, human resource concerns, and others.

The variety of the dataset allows for its use in testing intelligent email management systems due to the presence of real-life examples of business correspondence rather than text that was artificially created.

Even though there are over 500 thousand emails in the original dataset, several pre-processing steps had to be taken to make the dataset suitable for deep learning models. The redundant emails, the incomplete records, and very short emails were filtered out in order to make the dataset cleaner and ready for further processing.

Table (3.2): Characteristics of the Enron Email Dataset

Attribute	Description
Dataset Name	Enron Email Dataset
Data Type	Real corporate email messages
Language	English
Original Dataset Size	Approximately 517,000 emails
Final Cleaned Dataset	190,280 emails
Input Feature	Cleaned email text (Subject + Body)
Category Labels	Administration, Financial, HR, Legal, Market, Other
Priority Labels	High, Normal, Low

Key features of the dataset utilized throughout this work are presented in Table 3.2. The result of the data preprocessing procedure was 190,280 emails of high quality, which were then labeled and split into training, validation, and holdout sets.

The following figure depicts the process of preparing the dataset from the initial Enron dataset to the cleaned one used for modeling.

Figure 3.2: Dataset Preparation Pipeline

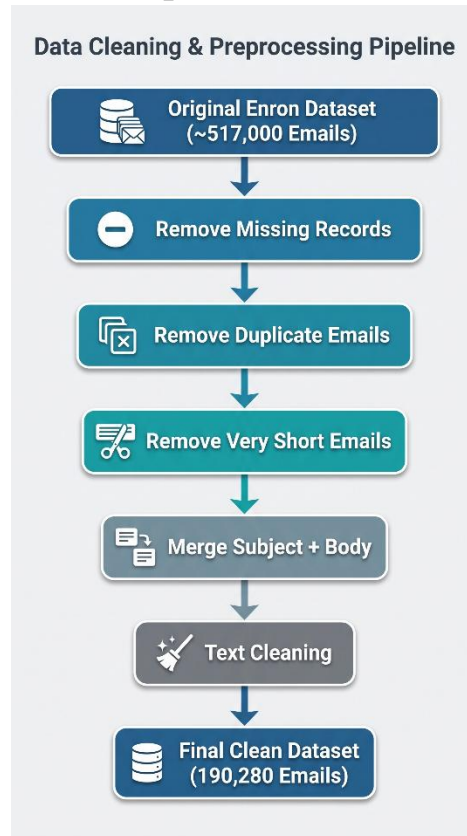


Figure 3.2 presents the primary pipeline of data preprocessing performed on the initial dataset from the Enron case study. Each step was specifically crafted to increase the quality of the data before automatic classification.

Such data preprocessing allowed eliminating noise and redundancies from the initial dataset, making it appropriate for both machine learning techniques and the novel DistilBERT multi-task model.

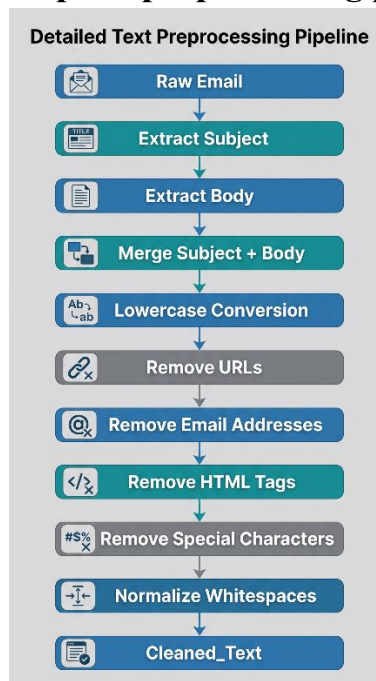
3.4 Data Preprocessing

To use the original dataset for training and testing, various preprocessing procedures had to be performed. As the original dataset has duplicate messages that are duplicates, incomplete messages, and emails with little textual content, a thorough preprocessing pipeline has been developed to ensure quality data for labeling and training.

Preprocessing involved noise reduction while retaining the semantic content of each email message. These processing operations ensured consistency between the

machine learning models and the DistilBERT multi-task model since they all use the same cleaned dataset.

Figure (3.3) illustrates the complete preprocessing pipeline applied in this study



The process is illustrated in Figure 3.3 as applied to every email before classification and model training.

After concatenation of the subject and body of each email into one text feature, multiple text preprocessing steps have been carried out. This procedure ensured removal of unnecessary formatting and transformation of the text to a consistent format while maintaining its semantics. The preprocessed text was used as an input for the baseline and proposed models.

3.4.1 Data Cleaning

Data cleaning was done with the objective of deleting entries that might affect the training process in any way. Entries with duplicate emails were deleted since they would create repetitions in the training sample, and those that had little textual information were deleted because they did not offer any valuable semantics to the classification.

Messages that had very little information content were deleted as well since they were usually signatures and greetings.

Table 3.3 summarizes the data cleaning process.

Cleaning Step	Description
Remove duplicate emails	Eliminated duplicate email records
Remove missing values	Removed incomplete records
Remove short emails	Excluded emails containing fewer than five words
Merge Subject and Body	Combined both fields into a unified text
Text normalization	Removed URLs, HTML tags, punctuation, and unnecessary spaces
Final cleaned dataset	190,280 email messages

Table 3.3 provides an overview of the pre-processing techniques used on the dataset before labelling and model building.

Pre-processing techniques helped achieve data consistency, decreased redundancy, and generated a good-quality dataset for machine and deep learning experiments.

3.4.2 Text Normalization

After completing the cleaning stage, the rest of the emails were subjected to normalization to create a uniform textual representation of the whole dataset. This normalization procedure minimized the variations in formatting while retaining linguistic information that was needed for the classification process.

Some of the common normalization steps involved:

- All text was changed to lowercase.
- All HTML tags were removed.
- Email and URLs were stripped out.
- Some punctuation and special characters were removed.
- Extra white space was also removed.
- The email subject and body were merged to form one text field.

Figure (3.4): Example of Text Preprocessing

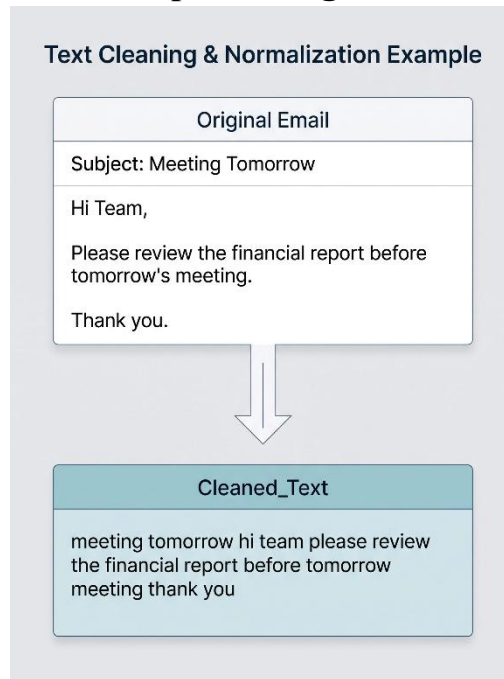


Figure 3.4 presents a sample demonstrating how raw email is converted into normalized text suitable for model input.

3.5 Data Labeling Strategy

Since the Enron Email Dataset provides no pre-defined labels for either email categories or priorities, it was necessary to develop an automatic labeling scheme for generating the target variables needed for supervised learning.

The labeling procedure included two separate sub-tasks. According to the first sub-task, each email received one or several functional categories, whereas the second sub-task attributed one priority level. Such an approach allowed building a multi-task learning dataset, which could perform not only category classification but also priority prediction. Figure 3.5 presents the general approach used to label the emails.

Figure (3.5): Automatic Labeling Workflow.

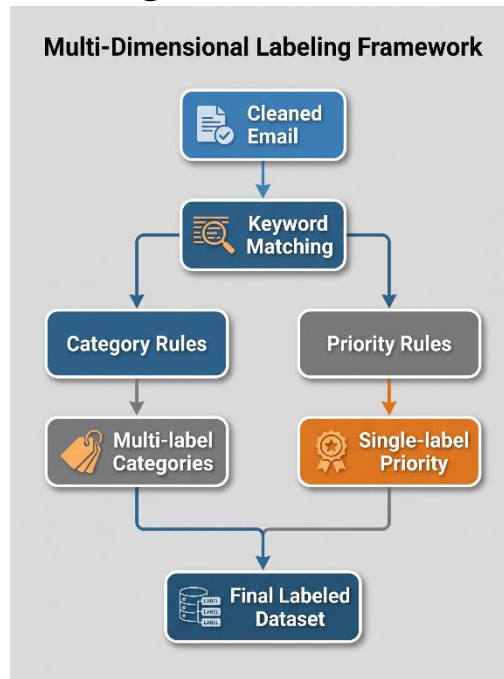


Figure 3.5 depicts the automatic labeling procedure used to generate the labels for the processed emails.

The labeling procedure was conducted according to the rule-based approach using domain-specific keyword dictionaries. After preprocessing, each email message was analyzed, and the appropriate keyword sets were identified, thus determining category labels and the priority level of the message.

Unlike a manual labeling procedure, such an automated approach allowed for labeling the email dataset efficiently and consistently.

3.5.1 Category Labeling

The process of classifying emails was modeled as a multi-label classification task since an email could be talking about more than one topic at a time.

Six categories were defined according to the goals of the suggested email management system:

- Administration
- Financial

- Human Resource (HR)
- Legal
- Market
- Other

Each of these categories was defined by a set of keywords and phrases. During labeling, an email would get a label for each category in which the corresponding keywords appeared in its clean text.

Table (3.4): Category Labels Used in the Study

Category	Description
Administration	Administrative requests, meetings, approvals, scheduling
Financial	Budget, invoices, payments, accounting, expenses
HR	Recruitment, leave requests, employee matters
Legal	Contracts, regulations, compliance, legal issues
Market	Marketing, customers, sales, campaigns
Other	Emails not matching any predefined category

Table 3.4 below shows the six functional categories utilized in this study. These categories were chosen to reflect common issues regarding organizational communications that occur in enterprise email services.

3.5.2 Priority Labeling

As opposed to the category prediction, priority prediction was considered as a single-label multiclass classification problem since one email should be assigned only one priority.

There were three priority levels created:

- High
- Normal
- Low

The priority assignment was done according to the predetermined list of urgent keywords and phrases. Email messages with any signs of urgency or need for quick action were assigned to the High priority level. If the email had no signs of urgency

but it was a routine business communication, then the Normal priority was assigned; otherwise, the Low priority was assigned.

Table (3.5): Priority Levels

Priority	Description
High	Immediate attention required
Normal	Routine business communication
Low	Informational or non-urgent messages

Table 3.5 illustrates the three priority levels used in the automatic labeling process. These labels were used as the target classes for priority classification.

3.5.3 Label Distribution

Once the automatic labeling step is performed, each email message will have the following two target variables:

- Multi-label category vector.
- Priority label.

The labeled dataset obtained during this step was used as a basis for further experiments and consisted of baseline machine learning models, the proposed multi-task DistilBERT model, threshold adjustment, and the results of final evaluation described in Chapter 4.

Figure (3.6): Structure of the Final Labeled Dataset

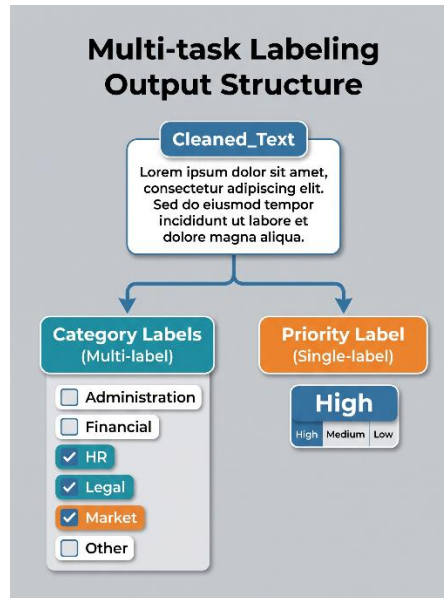


Figure 3.6 demonstrates the structure of the labeled dataset, which includes the input text and the generated category and priority labels.

Distribution of Category and Priority Labels

Once the automatic labeling step is completed, the distribution of the category labels is studied to get an idea about the balance of the dataset before its training. As the current research deals with a multi-label classification task, a single email message can be assigned to several categories. That is why the overall quantity of the category labels is larger than the quantity of email messages. The distribution of the category labels is demonstrated in Table 3.6.

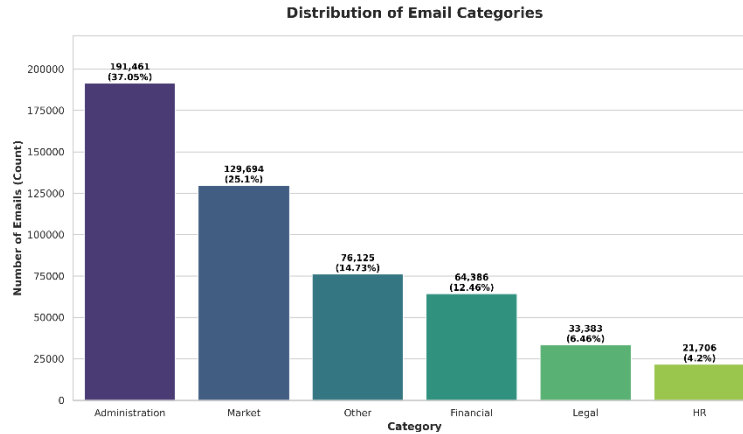
Table (3.6): Category Distribution

Category	Count	Percentage
Administration	191,461	37.05%
Market	129,694	25.10%
Other	76,125	14.73%
Financial	64,386	12.46%
Legal	33,383	6.46%
HR	21,706	4.20%

As seen from Table 3.6, the largest number of emails belongs to the Administration category (37.05%), and the second place is held by Market (25.10%). At the same time, categories HR and Legal have many fewer emails in them and make up 4.20% and 6.46% of all category labels, respectively.

These proportions demonstrate that our dataset is naturally imbalanced and does not contain the same amount of data for different classes. It demonstrates the features of natural data rather than a balanced dataset.

Figure (3.7): Distribution of Category Labels



This class imbalance can be observed in Figure 3.7, where the distribution of category labels after the automatic labeling process is demonstrated. The figure demonstrates the differences in category frequencies, with Administration being the most numerous and HR the least numerous among them.

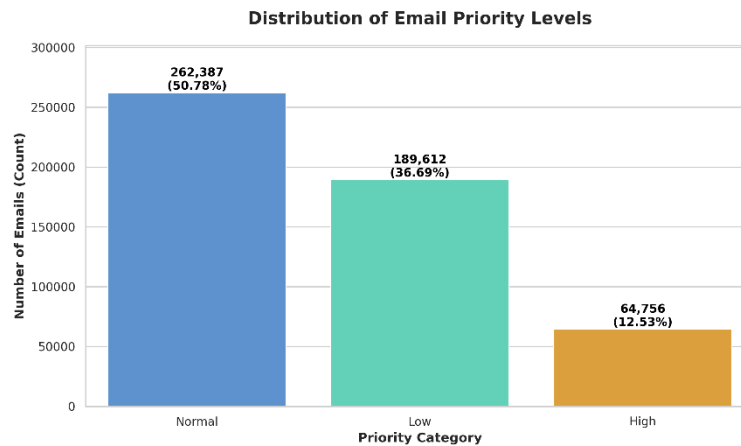
Besides the distribution of category labels, the distribution of priority labels was analyzed to see what proportion of emails had High, Normal, and Low priority levels. As priority is considered a one-label classification problem, only one label can be applied to each email. The distribution of priority labels is given in Table 3.7.

Table (3.7): Priority Distribution

Priority	Count	Percentage
Normal	262,387	50.78%
Low	189,612	36.69%
High	64,756	12.53%

Table 3.7 presents the distribution of priorities in the labeled emails dataset. Most emails in the dataset have a priority level of Normal, while the fraction of High priority emails is quite small, reflecting the inherent imbalanced nature of the priority classification problem and making evaluation metrics such as Macro F1-score and High-priority Recall important besides a simple accuracy measure.

Figure (3.8): Distribution of Priority Labels



3.6 Dataset Splitting

After performing the labeling task automatically, the resulting clean dataset is ready for building the models. To save computational power and still work with representative data, we took a stratified sample of 50,000 emails from the resulting clean dataset.

Further on, we split our dataset into three independent datasets to be used for building the model, optimizing its hyperparameters, and evaluation.

In particular, we used the stratified sampling method to maintain the distributions of both category and priority labels in all subsets.

Our dataset was split as follows:

- Training set (70%)
- Validation set (10%)
- Test holdout set (20%)

Such an approach to splitting our dataset guarantees that the test data is not seen during the model training nor hyperparameters optimization.

Table (3.9): Dataset Partitioning

Dataset	Number of Emails	Percentage	Purpose
Training Set	35,000	70%	Model training
Validation Set	5,000	10%	Model selection and threshold tuning
Holdout Test Set	10,000	20%	Final model evaluation

Total Sample	50,000	100%	Experimental dataset
---------------------	---------------	-------------	-----------------------------

Table 3.9 gives the final distribution of the experimental dataset that was used in the entire experiment.

The validation dataset was kept aside for model selection and optimization, while the test dataset was totally independent and used only once for the final evaluation of the proposed model.

Figure (3.9): Dataset Partitioning Strategy

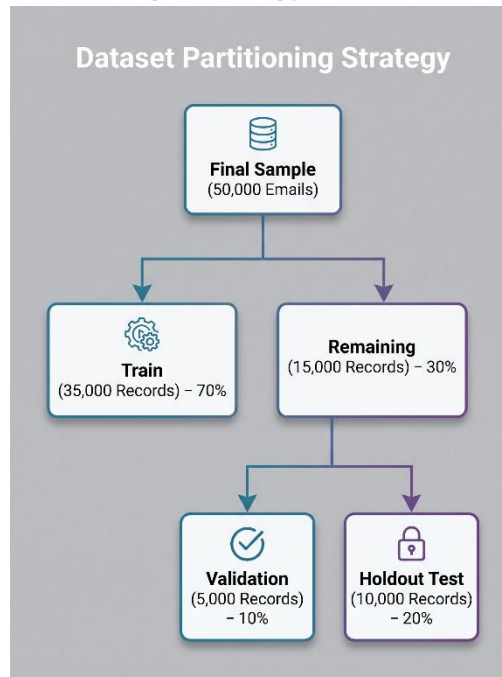


Figure 3.9 presents the adopted dataset partitioning approach

3.6.1 Stratified Sampling

As the dataset has an imbalanced distribution of classes for both the categories and the priorities, a simple random partition might create subsets with an inconsistent distribution of labels. To counter this problem, stratified sampling was used while splitting the dataset.

In the case of stratified sampling, the process was done based on Category-Priority labels, which would help to maintain the distribution of both prediction tasks within the training, validation, and holdout sets. This way, the sampling was unbiased and reliable.

The similarity of the distribution of labels within the three subsets is helpful for the comparative analysis between the baseline models and the DistilBERT multi-task model, since all experiments were done using the same data partitions.

Figure (3.10): Stratified Sampling Concept

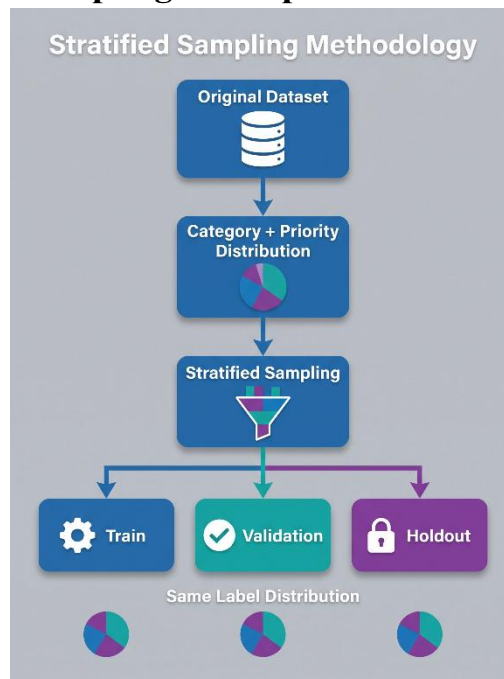


Figure 3.10 shows the sampling strategy that was used to maintain the distribution of the categories and priorities across all partitions of the dataset.

3.7 Baseline Models

Two traditional machine learning models were developed before the development of the proposed deep learning model in order to set baselines to compare the proposed model against.

Both baseline models were trained on the same training dataset and were evaluated under the same experimental settings that have been used for all experiments throughout this research. This ensures that any difference in the performance is due to the type of learning approach used and not due to differences in experimental settings.

Both the baseline models utilized the TF-IDF (Term Frequency–Inverse Document Frequency) representation to represent email texts as feature vectors for the machine learning algorithms.

Figure (3.11): Presenting the overall baseline modeling workflow.

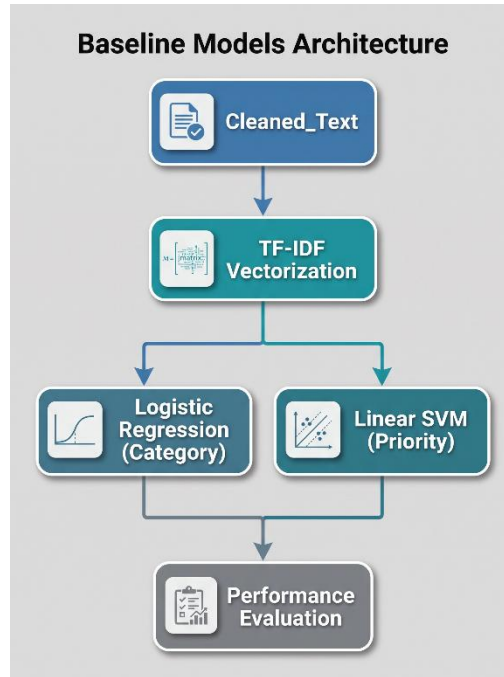


Figure 3.11 represents the workflows of the machine learning models used as baselines for comparison with the proposed DistilBERT model.

3.7.1 Text Representation Using TF-IDF

The baseline models made use of the TF-IDF feature representation method to convert the textual data to numerical feature vectors. TF-IDF calculates weights for each word in relation to its importance in a particular document compared to the whole dataset. Frequent words in the dataset are assigned low weights while more informative words gain high weights.

To ensure consistency across the experiments, only the 20,000 most informative words in the training dataset vocabulary were considered in TF-IDF representation.

Table (3.10): TF-IDF Configuration

Parameter	Value
Representation Method	TF-IDF
Maximum Vocabulary Size	20,000
Input Text	Cleaned Text
Output	Sparse Feature Vectors

Table 3.10 describes the configuration of the TF-IDF used in the baseline model.

3.7.2 Logistic Regression for Category Classification

Logistic Regression was the first baseline model that was used to complete the task of email categorization.

As the emails can belong to more than one category, the problem of categorization was treated as a multilabel classification problem. The approach of “one-vs-rest” was followed, which meant that a separate binary classifier could be trained for each category.

These classifiers gave the probability of the belonging of the email in their corresponding category, and finally the categories were decided using some decision threshold.

Figure (3.12): illustrates the classification pipeline.

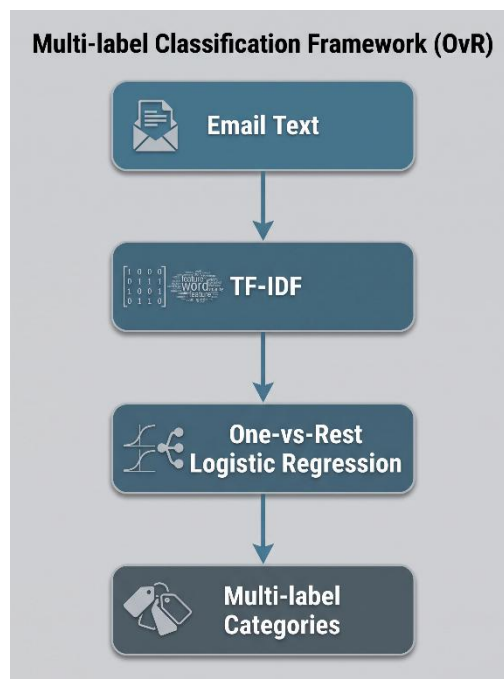


Figure 3.12 depicts the logistic regression approach for multilabel email category classification.

3.7.3 Linear SVM for Priority Classification

The second baseline model was the Linear Support Vector Machine (Linear SVM). Linear SVM was used to perform the task of priority prediction.

While category classification was a multi-label multi-class problem, priority prediction is a single-label multiclass problem since emails belong to only one priority category each. Linear SVM uses the TF-IDF feature extraction to determine the best possible hyperplanes between priority categories.

It was chosen due to its high effectiveness when applied to high-dimensional text classification tasks and computational efficiency.

Figure (3.13): illustrates the priority classification pipeline

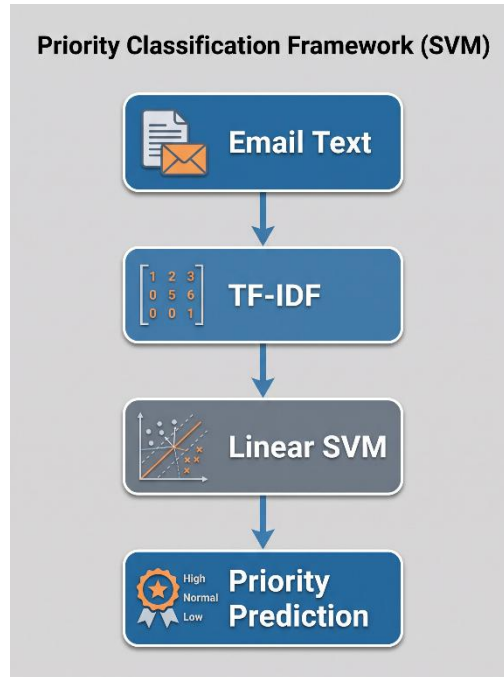


Figure 3.13 shows the linear SVM workflow for email priority classification.

3.7.4 Experimental Settings

Both baseline models were trained on the same training dataset and tested on the validation and holdout datasets mentioned above. The preprocessing pipeline and performance metrics were applied identically to both baseline models to facilitate a fair comparison with the DistilBERT model proposed in the next chapter.

Baseline experiments provided a baseline benchmark for evaluating the performance of the proposed model in Chapter 4.

Figure (3.14): Overall Research Methodology

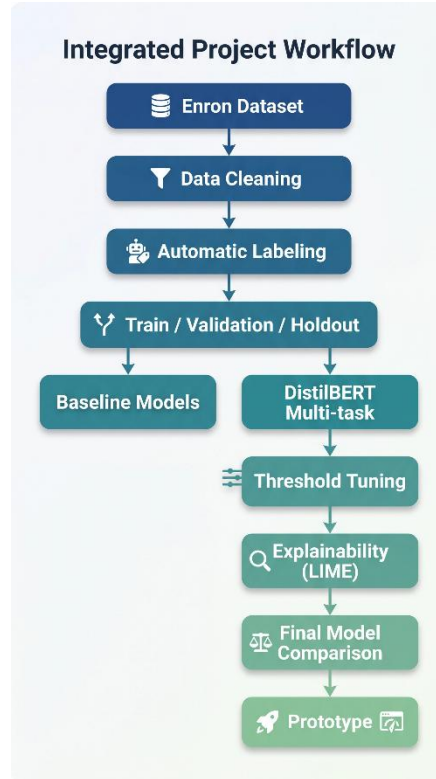


Figure 3.14 demonstrates the full research methodology applied in this study, starting from dataset preparation and preprocessing to automatic labeling, dataset splitting, the implementation of baseline and proposed DistilBERT models, threshold optimization, interpretability with LIME, model comparison, and prototyping.

Chapter 4: Implementation and Results

4.1 Introduction to the Chapter

In this chapter, the stages of the practical implementation of the study will be described, beginning with the preparation of the working environment and data loading, including the development of the baseline models and the development and testing of the proposed model based on DistilBERT.

In this chapter, the results of the training and testing will be discussed, including the analysis of the results and their comparative assessment to evaluate the ability of the proposed model to outperform traditional models in the classification of emails and determining their priorities.

All experiments were carried out using the prepared dataset, and the same training, validation, and testing datasets were used for all models to provide comparable results. All experiments were performed in an identical software environment and under identical training conditions to exclude the influence of execution conditions on the results.

The implementation process involved two distinct phases. The first phase involved developing baseline models through traditional machine learning methods: Logistic Regression for the classification problem and Linear SVM for the prioritization problem. The second phase involved developing the proposed model through DistilBERT following a multi-task learning method so that the model would be able to perform both tasks at once.

As part of the thorough evaluation of the model, this chapter does not confine itself to showing only the training results but also evaluates the performance through the use of a holdout test set, conducts threshold tuning to enhance the results in multi-class classification, and explains the model decisions using Explainable AI (LIME). The chapter ends with a comparison of all the models developed together with results analysis and shows how the proposed model has accomplished the objectives of the study.

4.2 Experimental Environment

To ensure efficient performance of all experiments, an environment for developing and training deep learning models and processing large amounts of text data was selected. All steps of the study were done in the Python programming language

because of the availability of special libraries for natural language processing, machine learning, and deep learning.

In the current study, the Google Colab platform was used as the main implementation environment because it allowed graphics processing units (GPUs), which allowed faster training of the DistilBERT model than the implementation on a CPU.

Table (4.1): Libraries Used in the Study

Usage	Library
Data loading and processing	Pandas
Numerical Operations	NumPy
Implementing Baseline Models and Evaluation	Scikit-learn
Loading the DistilBERT Model and Tokenizer	Transformers
Building and Training the Model	PyTorch
Plotting Training Curves and Results	Matplotlib
Explaining Model Decisions (Explainable AI)	LIME

Table 4.1 demonstrates that the project implementation involved several software libraries that served certain purposes during the implementation steps.

I have also taken advantage of the possibility of parallel processing offered by PyTorch and mixed-precision training to decrease the training time and optimize the usage of GPU resources without deterioration of the model quality.

To ensure reproducibility of the research, the best trained model was saved together with the tokenizer files, class labels, and priorities. Thus, the model will be easy to load in the future and use during the testing stage or in the application designed to show the results of the model.

4.3 Data Preparation for Implementation

The actual implementation part started after all data preparation had been completed and the train, validation, and test sets created. All the models developed for the study made use of these files to make sure that all comparisons were based on the same dataset.

First, the Train Set, Validation Set, and Holdout Test Set were loaded, and the data checked for integrity based on the number of samples and column names, and any missing data that might interfere with the training process.

The next step involved preprocessing the data based on the kind of model. For the baseline models, the final column of text (Cleaned_Text) was used for the extraction of TF-IDF features. However, in the case of the DistilBERT model, the same column was used for the conversion of text to token IDs and attention masks using the Tokenizer.

In addition to that, all labels related to each message, including multiple-class labels (Category Labels) and priority labels (Priority Label), were also loaded for their usage in future training and testing of the different models.

With the conclusion of this process, the data was now set for the implementation of baseline models, which are the reference point for our model discussed further ahead in the paper.

4.4 Implementation of Baseline Models

4.4.1 Introduction

The base models marked the first stage of practical experiments performed within this study, where the objective was to develop the benchmark on whose basis the performance of the suggested model would be evaluated.

For this purpose, two classic machine learning approaches were employed: the Logistic Regression approach was used to solve the problem of category classification, while the Linear Support Vector Machine (Linear SVM) approach was used for solving the priority classification problem.

Both models made use of the same dataset created in the previous chapter, as well as the same train, validation, and test sets.

4.4.2 Feature Extraction Using TF-IDF

To train the baseline models, it was required to convert the textual data into a numerical format, which can be used by machine learning algorithms.

For this purpose, the TF-IDF Vectorizer technique was applied to find the features from the (Cleaned_Text) column, where each message is represented in the form of a numeric vector according to the importance of words within the message compared to all other messages in the dataset.

In the process of implementation, the training dataset was used for finding the vocabulary, and then the same vocabulary was applied to the validation and test datasets in order not to allow any leakage of the test dataset to the training stage.

The same parameters were used for all experiments to make sure that all baseline models have been built using the same text representation.

4.4.3 Training the Logistic Regression Model

After the extraction of the TF-IDF features, I started to train the Logistic Regression model to perform the class classification task.

As a message can belong to several classes, the model was applied in the framework of One-vs-Rest (OvR), meaning that the model was trained separately for each of six classes, and then results from all models were aggregated to get the final classification for each email.

During training, the model was used only with the training set, while the validation set was used to measure the performance of the model before the final evaluation stage with the test set.

Finally, after the training of the model, predictions for test data were made, and the calculation of all performance metrics for the class classification task was conducted.

4.4.4 Results of the Logistic Regression Model

The results of the evaluation of the model proved that the model worked effectively for the category classification task, which is proof of the efficiency of the combination of TF-IDF and logistic regression in the identification of general patterns within emails. The table below demonstrates the results of the final evaluation of the model.

Table (4.2): Logistic Regression Results for Category Classification

Metric	Value
Subset Accuracy	0.6840
Macro F1-score	0.8751
Micro F1-score	0.8711
Hamming Loss	0.0691

Jaccard Score	0.8280
---------------	---------------

Analysis of Results

It could be noted that the model has quite a high Macro F1 score, indicating the possibility of the model classifying various categories with a sufficient degree of accuracy.

At the same time, the Subset Accuracy score is lower in comparison with other metrics, which is normal in the case of multi-label classification, since this metric treats the prediction as correct only when the model can predict all the categories belonging to a particular message without any mistakes.

The Hamming Loss indicator has also shown that the number of mistakes in the process of classifying individual labels was low; however, the fact that the model uses mainly the frequency of usage of words in the text without considering the context of their usage is one of the major weaknesses of such models.

Despite the successful results, the model under discussion is used in the current paper as a benchmark in the comparison with the proposed model.

4.4.5 Summary of Phase Implementation

This step is an initial experiment performed during the course of the study where text vectors are created using the TF-IDF method and a Logistic Regression model is trained on multi-class classification and evaluated on the test set.

This result offers a baseline that can be used later to compare the improvement gained through the use of the proposed model, which will be further tested in the next part of this research through the creation of a Linear SVM model.

4.4.6 Implementation of the Linear SVM Model for Priority Level Determination

Following the assessment of the Logistic Regression Model to classify the classes, the next step was to implement the second baseline model called Linear Support Vector Machine (Linear SVM) to classify the priority level of the emails.

The linear support vector machine utilized the same data representation as the one employed in the previous experiment, thus allowing for the determination of the difference in performance of the algorithm itself and not in its data representation.

In the process of building the model, the training dataset was used first, then the validation set was applied to check the performance of the model, and finally, the Holdout Test Set was used to test the performance of the model.

As can be seen from above, the prioritization task performed in this research is a single-label multi-class classification problem where the algorithm should assign a message to one of three priority levels (High, Normal, or Low).

4.4.7 Results of the Linear SVM Model

When the training process was finished, the model was tested on the final test set, and all of the performance metrics of the task were measured. The following table presents the results gained by the model.

Table (4.3): Linear SVM Results for Priority Level Determination

Metric	Value
Accuracy	0.9079
Precision	0.9092
Recall	0.9079
Macro F1-score	0.8695

The evaluation did not consist only of overall metrics; the performance of the model was also analyzed on each priority level separately in order to have a better understanding of the model's advantages and disadvantages.

The analysis indicated that the model worked fine on the classification of messages of Normal and Low priority levels, but the performance dropped on the messages of High priority, especially in the Recall metric, which was about 0.69.

It means that the model could not find all high-priority messages, which is normal considering their number in comparison with other priority levels.

Analysis of Results

Based on the results, it is evident that the Linear SVM model has demonstrated a very good accuracy, which was above 90%, proving that the model works effectively when dealing with email data, which had been converted into TF-IDF format.

Nevertheless, a deeper analysis of the results shows that the performance of the model varied depending on the priority level. The model was able to provide stable results for the most frequent classes in the dataset; however, its capacity to classify high-priority emails decreased, which is evidenced by the Macro F1-score.

It should be mentioned that this issue is connected not so much to the algorithm as to the nature of the dataset, as the “High” class was the least frequent among all classes. These results prove one of the key limitations of traditional models, which are based on the statistical features extracted from words and do not use any contextual information between message words.

4.4.8 Summary of the Results of the Baseline Models

Now, after implementing the two reference models, the study has a benchmarking model that will help to measure the performance of the proposed model. Below is the table that presents the results obtained from the reference models.

Table (4.4): Summary of Reference Model Results

Key Performance Indicators	Model	Task
Macro F1 = 0.8751 , Hamming Loss = 0.0691 , Subset Accuracy = 0.6840	Logistic Regression	Category Classification
Accuracy = 0.9079 , Macro F1 = 0.8695	Linear SVM	Priority

Stage Analysis

The performance results of the baseline models indicated that traditional machine learning algorithms have the capacity to perform very well on emails once they are processed using TF-IDF representation. However, the models still depend on the statistical properties of the texts without considering the linguistic context and the semantic relations between words. This might make it difficult for the model to handle emails that require a high-level understanding of the text.

Thus, in the next stage, the study proceeded to implement the proposed DistilBERT-based model to make use of the contextual representation of the text and multi-task learning to classify categories and rank priorities within a single model.

4.5 Implementation of the Proposed DistilBERT Model

4.5.1 Introduction

Once the implementations of the baseline models and the evaluation of their performance were finished, it was time to implement the proposed model based on DistilBERT, which is the primary contribution of this work.

In contrast to the baseline models, which used feature extraction with TF-IDF and then trained separate models for each task, the proposed model used a single language model, which could extract features from the text automatically and classify categories and prioritize at the same time using multi-task learning.

The training, validation, and test datasets used by the baseline models were utilized to make sure that the comparisons between all models were fair.

4.5.2 Data Preprocessing for the Model

Before the start of the training process, preprocessing of the data was conducted in line with the DistilBERT model requirements. The model considered the (Cleaned_Text) column as the input (text) data for training, while category labels and priority levels were taken as target variables (labels).

Since the Transformer models cannot work directly with text data, the DistilBERT Tokenizer was applied for conversion of each message into its numeric form, which consists of:

- Token IDs.
- Attention mask.

Also, the maximum length of a message (Maximum Sequence Length) was defined so that all messages would have the same length inside each batch of training data; the shorter messages would be padded, and the longer ones truncated.

After the tokenization step was finished, the data was converted into PyTorch Dataset objects and then into a DataLoader.

4.5.3 Model Training

After the data had been prepared, the model was developed based on the Transformers library. The model used a pre-trained DistilBERT model as a common feature extraction module and two separate output modules for performing the two tasks.

The first output module was used for the category classification task, providing the probability of belonging to one of the six categories, and the second output module was used for the priority level determination task, providing the probability of belonging to one of the three priority levels.

Thus, the model is capable of processing the message only once and then providing results of both tasks at once.

4.5.4 Preparing the Training Process

After completing the modeling part, everything needed to train the model was ready. This phase involved:

- Defining loss functions for each task.
- Choosing an optimizer.
- Defining the learning rate.
- Defining the batch size.
- Number of epochs for training.
- Enabling training through GPU (Graphics Processing Unit).

Mixed Precision Training was also applied to minimize memory usage and improve the training time while keeping the stability of the weights during the training phase. To keep the best version of the model, I kept the model that performed best on the validation data while training and used it for all evaluation processes.

4.5.5 Training Hyperparameters

Training relied on a set of fixed settings, which were used throughout the model fine-tuning process.

Table (4.5): Training Settings for the Proposed Model

Parameter	Value
Pre-trained Model	distilbert-base-uncased
Framework	PyTorch (v2.x)
Transformers Version	4.x
Tokenizer	DistilBERT Tokenizer
Maximum Sequence Length	256 tokens
Truncation Strategy	Truncate from end
Padding	Pad to max_length
Batch Size	16
Number of Epochs	5
Learning Rate	2×10^{-5}
Optimizer	AdamW
Optimizer Epsilon	1e-8
Weight Decay	0.01
Scheduler	Linear LR Scheduler
Warmup Ratio	0.1
Dropout	0.1
Category Loss	BCEWithLogitsLoss
Priority Loss	Cross-Entropy Loss
Loss Weights	0.5 (category), 0.5 (priority)
Multi-label Threshold	0.5
Mixed Precision	Enabled
Early Stopping	Best validation model
Random Seed	42
Trainable Parameters	~67M
Model Type	Multi-task Learning
Category Output	Multi-label (6 classes)
Priority Output	Single-label (3 classes)
Hardware	NVIDIA CUDA GPU

Phase Analysis

This step is the start of the implementation of the proposed model, as the experiment stopped using the classical feature extraction approach and turned to using a pre-

trained language model which could learn context representation of texts directly. In addition, the application of multi-task learning in the training helped perform category classification and priority assignment simultaneously, using the same data as in the case of baseline models.

These settings have been necessary to prepare the model for fine-tuning by updating its parameters with the help of data collected during the experiment, while monitoring its performance on the validation set, until reaching the optimal model – as will be discussed in the next chapter.

4.5.6 Model Training

Once the model configuration and dataset preparation phases had been completed, the fine-tuning stage started for the DistilBERT model on the training dataset, with validation accuracy calculated after every epoch completion on the validation set.

During the fine-tuning stage, the weights of the model were updated iteratively based on the batch processing of the data; the text was processed in the form of token IDs and attention masks, whereas category classifications and priorities were used as ground truths.

The sum of the losses obtained by computing the loss function on the category classification problem and the loss function on the prioritization problem in each epoch constituted the total loss value, which was used to update the model's weights. Every time after the completion of an epoch, the model was assessed using the validation set, and performance measures were computed to monitor its performance and choose the best version before the testing stage.

4.5.7 Monitoring the Training Process

However, in order to assess the training progress, the loss value was not the only indicator considered; at the end of each epoch, the performance indicators relevant to each task were calculated.

For the class classification problem, the values of Macro F1 score and Hamming Loss were calculated, whereas for the prioritization problem, Accuracy and Macro F1 score were calculated.

This way, the performance of the model was monitored more extensively, ensuring that a decrease in the loss value would be accompanied by an increase in the

classification performance. Also, these indicators helped to determine when the best point to save the model was.

4.5.8 Saving the Best Model

At the end of each training loop, the results of the model at that stage were evaluated against the best result so far obtained on the validation set.

If there was an improvement in the evaluation metrics, the model together with the tokenizer, classes, and priorities was saved such that it could be used in future without having to go through another training process.

All other experiments, whether testing on the test set or using the model in practice, were done using this saved model.

4.5.9 Training Results

The training process showed steady improvement in the performance of the model with the progression of each epoch. With the training of the model, the training loss continued to decrease, while the validation performance continued to increase in relation to both categories and priorities. This shows that the proposed model for multitask learning based on DistilBERT learned useful representations from the email data for performing two tasks at once through one shared encoder.

The best validation performance was obtained at the fourth epoch of the training process, when the model achieved its minimum validation loss, along with maximum validation performance. The values of training and validation metrics for this epoch are shown in Table 4.6.

Table (4.6): Training and Validation Performance Across Training Epochs

Epoch	Training Loss	Validation Loss	Category Macro F1	Priority Macro F1
1	0.6249	0.3340	0.9315	0.8919
2	0.2649	0.2317	0.9578	0.9217
3	0.1866	0.2195	0.9655	0.9206
4	0.1340	0.2051	0.9677	0.9350
5	0.0944	0.2458	0.9677	0.9314

4.5.10 Analysis of Training Results

From the results of the training, it can be said that the proposed model started to converge slowly but steadily throughout the training process. The fact that training loss is constantly decreasing along with the improvement in the validation metrics indicates that the model successfully captured the underlying patterns of a semantic nature that exist within the dataset of emails.

Besides that, the multitasking nature of the approach made it possible to make the DistilBERT encoder learn a common context representation that is used for category classification and prediction of priorities, which makes it unnecessary to build two separate deep learning models for solving these tasks separately.

Even though the training loss continuously decreases within the fifth epoch, the validation loss slightly increases from 0.2051 to 0.2458, and the Priority Macro F1-score decreases from 0.9350 to 0.9314. Thus, the signs of overfitting appear after the fourth epoch. The model of the fourth epoch was chosen as the best one.

4.6 Model Evaluation Using the Holdout Test Set

4.6.1 Introduction

Once the training phase was completed and the best checkpoint according to the validation phase was selected, the proposed model was tested on the independent Holdout Test Set.

Unlike the training and validation sets, the holdout set was not exposed at all in the training and threshold tuning processes. Therefore, the presented results can be considered unbiased since they reflect the model's generalization abilities when dealing with unknown emails.

The model from the best validation epoch was applied to make predictions on all test samples. Threshold tuning allowed obtaining the optimal threshold value, which was applied for making final class predictions based on the probabilities. Finally, performance metrics were computed separately for both category classification and priority classification tasks.

4.6.2 Evaluation of the Category Classification Task

The initial evaluation was related to the multi-label category classification task. The probability of belonging to each of the six predefined categories for every email from the holdout test dataset was obtained from the trained DistilBERT model.

Then the threshold values, found in the process of validation, were applied to the probability values to get the binary labels for each email. After that, the obtained predicted labels were compared with the corresponding ground-truth labels.

The next table shows the final results of the category classification task.

Table (4.7): Final Category Classification Performance on the Holdout Test Set

Metric	Value
Macro F1-score	0.9672
Hamming Loss	0.0175
Macro Precision	0.99
Macro Recall	0.95
Optimized Threshold	0.60

Analysis of Results

From the experiment, it is obvious that the proposed multi-task DistilBERT model performs very well on the category classification task. The obtained Macro F1-score is equal to 0.9672 and is accompanied by a very low Hamming Loss of 0.0175. Thus, the model makes multi-label predictions with extremely high accuracy and almost no errors.

This result confirms the efficiency of deep language representations in the case of capturing semantic relationships in emails. In comparison with the machine learning baseline models discussed in this study earlier, the proposed model gives more accurate classification results.

4.6.3 Evaluation of the Priority Classification Task

The second evaluation was conducted to measure the performance of the multi-task model on the priority classification task. Since priority classification is a one-to-one problem, the model classified each email based on the predicted priority class that had the maximum probability. The predicted classes were compared to the ground-

truth labels to calculate the evaluation metrics. The results for the priority classification task are summarized in Table 4.8.

Table (4.8): Final Priority Classification Performance on the Holdout Test Set

Metric	Value
Accuracy	0.9529
Macro F1-score	0.9327
Macro Precision	0.94
Macro Recall	0.93
Recall for High-Priority	0.8627

Analysis of Results

The designed multi-task DistilBERT model also achieved good performance on the priority classification task. The model achieved an Accuracy score of 0.9529 and a Macro F1-score of 0.9327, reflecting equal performance in predicting emails across all priority classes.

What is even more important, the model achieved a high Recall of 0.8627 for the High priority class, reflecting the ability of the model to correctly detect emails that required urgent attention without the risk of missing any such messages.

4.6.4 Overall Performance Analysis

The last evaluation proves the efficiency of the proposed multi-task DistilBERT model in terms of performing both tasks (category classification and priority prediction) via a single unified model.

Indeed, the multi-task learning approach proves its efficiency in such a way that a single encoder is able to learn the context representation beneficial for both tasks. At the same time, when compared to traditional TF-IDF-based baseline models proposed in the previous sections of this work, the proposed approach performed better in almost all metrics of evaluation.

Finally, integration of the two tasks into a single model leads not only to decreased computational complexity but also does not influence the model's predictive capacity.

4.6.5 Test Phase Results

The results shown in this section are the final results of the model evaluation on the holdout test set using the optimized threshold from the previous section of this work.

The results obtained can be considered as the benchmark results of the model generalization ability and will be used for comparative analysis later in this work.

Thus, testing the model on new unseen data is used for the objective evaluation of its performance for real-world tasks.

4.7 Threshold Tuning

4.7.1 Introduction

During the validation phase, it was found that the performance of the proposed classifier depends on the threshold applied in classifying the predicted probabilities.

When applying multi-label classifiers, the classifier does not classify. The classifier generates probability values independently for each category. The threshold comes in handy in determining which category will be allocated to the email based on the predicted probabilities. Any category whose predicted probabilities are above or equal to the threshold is allocated, while any other category below the threshold is discarded.

At first, the default threshold value of 0.50 was used. The default threshold may not yield the best performance for all the datasets. A process of threshold tuning was carried out to find the best threshold.

4.7.2 Threshold Tuning Methodology

Threshold tuning was carried out after the training of the DistilBERT model based on the checkpoint providing the best results in terms of the validation dataset. The aim was to find the optimal threshold without changing the trained model itself or its architecture.

The thresholds were checked incrementally, and the performance was compared for each value of the threshold. The optimal threshold was chosen based on two performance metrics:

- Macro F1-score
- Hamming Loss

These metrics were chosen due to their better reliability in evaluating the performance of the multilabel classifier compared to accuracy.

4.7.3 Experimental Results

The experiments showed that varying the threshold led to considerable changes in classification results. The lower values of the threshold increased the number of labels to predict, which improved the ability to detect classes, but increased the

number of false positives. On the other hand, the increase in the threshold values reduced errors in prediction but also led to higher chances of missing some relevant classes.

As a result of testing all possible threshold values, the threshold for the best classification of the categories turned out to be:

Threshold = 0.60. This threshold was then used for the final evaluation of the proposed model.

Table (4.9): Optimal Threshold Values Selected for the DistilBERT Multi-task Model

Task	Threshold	Best Metric	Score
Multi-label Category Classification	0.60	Macro F1-score	0.9681
Multi-label Category Classification	0.60	Hamming Loss	0.0171
Priority Classification (High Priority)	0.50	High Recall	0.8571
Priority Classification (High Priority)	0.50	Macro F1-score	0.9350

The tuning of the threshold was carried out using the validation dataset for choosing the best decision thresholds for both classification tasks. The threshold value of 0.60 produced the best results for multi-label category classification, leading to the Macro F1-score of 0.9681 and Hamming Loss of 0.0171.

As for priority classification, the best performance with consideration of the balance between classification results and recall of the High priority class was achieved with the use of the default threshold value (0.50), leading to the High Recall of 0.8571 and Macro F1-score of 0.9350.

4.7.4 Analysis of Results

Based on the experiment results, it is safe to conclude that the default threshold (0.50) was not the optimal one for the chosen dataset.

The increase in the threshold value up to 0.60 resulted in fewer mistakes in predicting category labels; as a consequence, there was a decrease in the Hamming Loss and an increase in the Macro F1-score value.

The reason for the improvement is that by increasing the threshold, the classifier became more conservative about making decisions about the categories. As a result,

only the predictions that had high confidence were considered. In other words, false positives were decreased without decreasing recall.

This means that choosing the optimal threshold is a significant step that needs to be taken to improve the performance of multi-label classifiers.

4.7.5 The Effect of Threshold Tuning on Model Performance

After threshold optimization, the model was again evaluated using the same validation dataset and the same experimental settings. It was found that there was a slight improvement in the performance of the classification task category.

It is noted that:

- There was a slight improvement in the value of Macro F1-score.
- There was a slight decrease in the value of Hamming Loss.

The model was not retrained or modified in any way to achieve these results. Rather, they were due to optimization of the decision threshold used to make decisions about the categories. As seen from Table 4.10, the increment of the threshold from 0.50 to 0.60 brought about a slight improvement in the value of Macro F1-score (from 0.9677 to 0.9681) and a slight decrease in the value of Hamming Loss (from 0.01713 to 0.01710).

Table (4.10): Effect of Threshold Tuning on Category Classification Performance

Metric	Before Threshold Tuning	After Threshold Tuning	Difference
Threshold	0.50 (Default)	0.60	+0.10
Macro F1-score	0.9677	0.9681	+0.0004
Hamming Loss	0.01713	0.01710	-0.00003

Section Analysis

At this point, it becomes clear that improving the model's efficiency does not always depend on re-training or changing the architecture of the model itself. There is an alternative approach which consists of improving performance through adjusting the parameters used at inference time, such as the decision threshold. As has been seen from the experiment, the optimization of the threshold led to some minor improvement in the quality of classification of categories without the need to change

the model itself. The minor improvement that has been achieved through this approach proves that it can be used effectively to increase prediction quality at low costs.

4.8 Explainable Artificial Intelligence (XAI)

4.8.1 Introduction

Even though the proposed DistilBERT multi-task model provided outstanding prediction results, transformer-based deep learning models are usually considered black-box models, as they give predictions but do not explain the reasoning process behind them.

To make the proposed system more transparent and trustworthy, the Explainable Artificial Intelligence (XAI) phase was included in this research. Instead of making improvements to the predictive ability of the model, the purpose of this phase was to explain how the trained model predicted and what textual features were influential in its decision-making.

One of the many available XAI methods, Local Interpretable Model-agnostic Explanations (LIME), was chosen because it explains the prediction on the local level without any changes to the trained model. LIME is model-agnostic and therefore can be used for deep learning models like DistilBERT.

4.8.2 Implementing LIME

After finishing the training process and choosing the model that showed the highest performance among the DistilBERT models, the LIME framework was used to provide explanations for the predictions made by the proposed model.

For each chosen email, the message was initially sent through the trained DistilBERT model to get prediction probabilities. The LIME algorithm produced multiple variations of the same message by selectively masking or deleting certain words in it. Then, each of the obtained variations was sent through the trained model to obtain new prediction probabilities.

Using these predictions, the LIME algorithm fit a local surrogate model around the initial email to make the contribution of each word to the prediction. Positive

contributions mean that the word positively affects the probability of the predicted class, while negative contributions mean the opposite.

To study the behavior of the proposed model and see how it operates, several email samples were chosen and tested using this approach. Figure 4.8 shows the entire pipeline of using LIME to explain predictions of the proposed DistilBERT model.

Figure (4.1): The workflow of the LIME explanation process for the proposed DistilBERT model

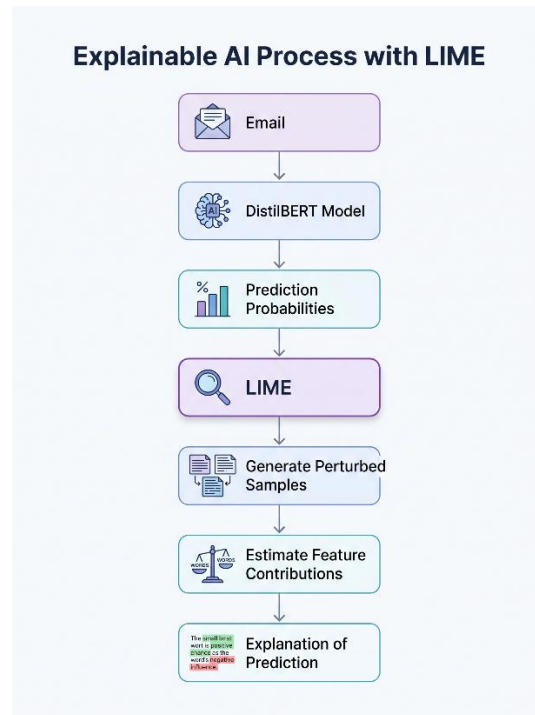


Figure 4.1 depicts the process of the LIME explanation workflow for the proposed DistilBERT model. It presents the steps of the Explainable Artificial Intelligence (XAI) workflow in order: from the initial input email to the prediction of the proposed DistilBERT model, generation of perturbed samples with LIME, calculation of feature (word) contribution, and finally, providing the visualization of the prediction explanation.

4.8.3 Interpretation Results

The obtained LIME explanations have shown that the proposed model depended mainly on semantically relevant words and domain-specific terminology, which were related to the email content.

For instance, words such as those related to finance, legal papers, meetings, agreements, human resources management, and urgent requests always positively influenced the respective prediction category. On the other hand, ordinary or context-independent words usually do not affect the final prediction in any way.

It was also clear from the explanations that the relevant words varied in different email examples, which means that the model used the contextual meaning of each email, but did not rely on a set of predefined keywords. This is an expected result since transformer-based models understand the context of language.

4.8.4 Analysis of LIME Results

LIME was very helpful in understanding how the model makes predictions based on the explanation of the words that contributed most significantly to the decision-making process.

The explanations that LIME generated generally pointed out that the model made decisions based on the relevant contextual meaning of the messages rather than arbitrary or irrelevant words. Such a finding shows that the model has actually learned semantic representations of the training data and has not made decisions based on lexical patterns only.

On top of that, examining various email messages has shown that the explanations generally matched the semantic context of the message itself. Thus, the model's predictions can be considered valid, and this finding shows that the proposed model can be applied to tasks requiring some level of interpretability.

At the same time, LIME also helped find out some cases when certain words had a disproportionately significant impact on the final decision. This information can be quite useful for diagnosis and further improvement of the model.

4.8.5 The Importance of Using XAI in the Study

The purpose of using LIME in this work was not to enhance the prediction ability of the proposed approach but rather to obtain interpretations of the predictions after training.

Using XAI in this work resulted in many benefits such as the following:

- Increased transparency of the proposed model.
- Explanation of the reasons for particular predictions.
- Demonstrating that the model relies primarily on meaningful contextual words.
- Enhancement of the reliability of experimental results.
- Analysis of both correct and wrong predictions.

In conclusion, the XAI helped to complement the quantitative analysis with a qualitative one by explaining how the proposed approach makes its predictions. This way, it increased user trust in the proposed approach and made it more appropriate for application.

Section Analysis

Despite DistilBERT being a black-box approach to deep learning, the use of LIME helped to provide local explanations, which increased the interpretability of the model in terms of making individual predictions. It was shown that the model based its predictions mainly on semantically meaningful words associated with the email content and priority instead of random and irrelevant features.

In addition, LIME served as a practical way to analyze both accurate and inaccurate predictions made by the model and therefore gave more insight into the behavior of the model beyond the metrics used in the performance evaluation. Hence, the XAI stage became another vital component of the suggested approach besides the evaluation of performance.

4.9 Final Model Comparison

4.9.1 Introduction

After conducting the baseline experiments, training the proposed multi-task model based on DistilBERT, tuning the decision threshold with the use of validation data, and testing the final model on an independent holdout test set, a detailed comparison of the proposed model and the baseline models performing best was performed.

The purpose of the comparison is to estimate the performance gain that is obtained by the proposed model under the same experimental conditions. Both the test dataset

and the evaluation metrics were the same for all models in order to make the comparison objective.

Considering that there are two different tasks in this study, the comparison is performed separately for category classification and priority classification. In addition to the comparison tables, there are visual representations which help to compare the performance of the proposed model and the baseline model.

4.9.2 Comparison of Category Classification Results

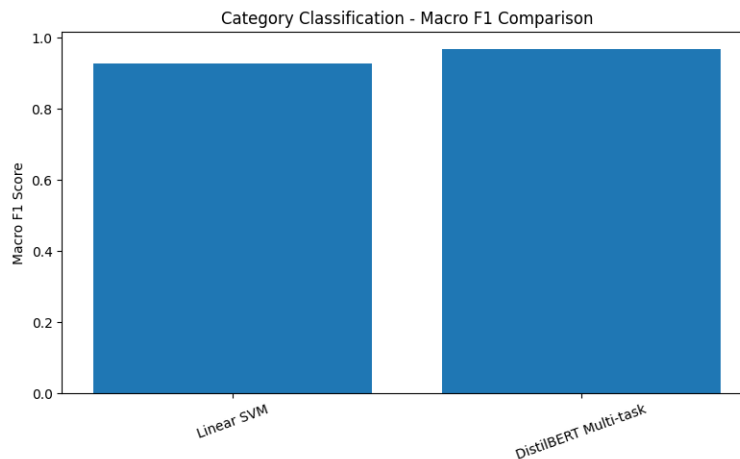
Table 4.11 demonstrates the performance comparison of the best traditional machine learning model (Linear SVM with TF-IDF features) and the proposed DistilBERT multi-task model on the category classification task.

Table (4.11): Comparison of Category Classification Results

Metric	Linear SVM	DistilBERT	Improvement
Macro F1-score	0.9269	0.9672	+0.0403
Hamming Loss	0.0370	0.0175	−0.0195

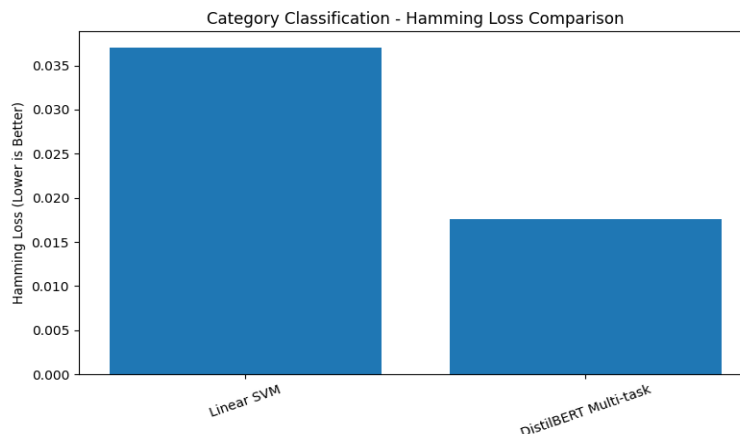
From the above table, it is evident that the proposed DistilBERT model performed better in terms of higher classification performance with fewer classification errors. Figure 4.2 below illustrates the improved Macro F1-score of the proposed model compared to that of Linear SVM, which indicates the superiority of the proposed model in terms of better email classification for all six categories.

Figure (4.2): Comparison of Category Classification Macro F1-score between Linear SVM and DistilBERT.



In the same way, Figure 4.3 below illustrates the improved Hamming Loss obtained by the proposed DistilBERT compared to that of Linear SVM, which indicates fewer erroneous labels during multi-label email classification.

Figure 4.3: Comparison of Hamming Loss for Category Classification.



Analysis of Results

The proposed DistilBERT model performed better than the best traditional baseline model on the email category classification task.

It improved the Macro F1-score from 0.9269 to 0.9672, indicating better classification performance in terms of balance in all six categories. In addition, the Hamming Loss was greatly reduced from 0.0370 to 0.0175, indicating fewer label-wise prediction errors.

This performance improvement is due to the contextual representation of languages through DistilBERT compared to the statistical representation through TF-IDF in Linear SVM.

4.9.3 Comparison of Prioritization Results

Table 4.12 presents the comparison between Linear SVM and the proposed DistilBERT model for the priority classification task.

Table (4.12): Comparison of Prioritization Results

Metric	Linear SVM	DistilBERT	Improvement
Accuracy	0.9079	0.9529	+0.0450
Macro F1-score	0.8695	0.9327	+0.0632

Recall (High Priority)	0.6900	0.8627	+0.1727
------------------------	--------	--------	---------

The superiority of the proposed model is further illustrated in the graphical comparisons.

Figure (4.4): Comparison of Priority Classification Macro F1-score.

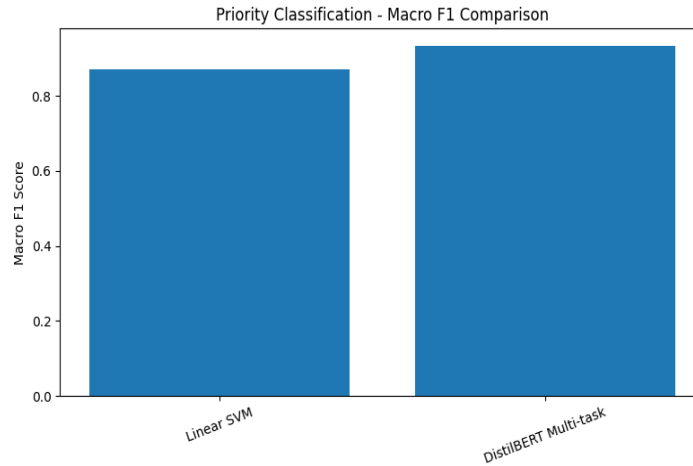
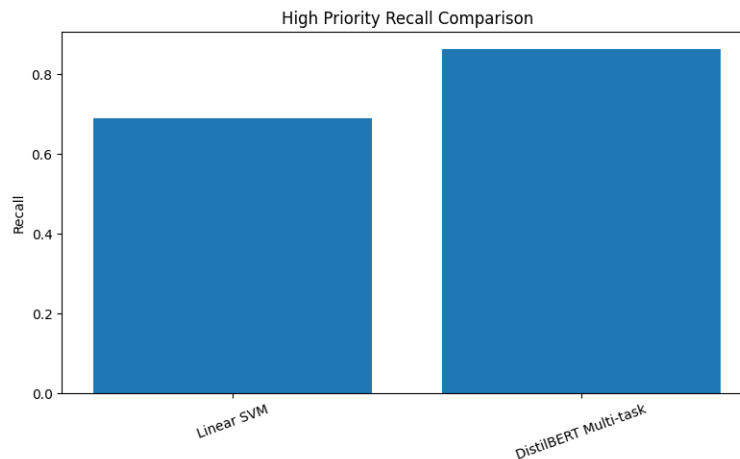


Figure 4.4 shows that DistilBERT obtained a better Macro F1-score in comparison to the Linear SVM, which implies a more balanced performance across all priority classes.

Moreover, as can be seen in Figure 4.5, DistilBERT also performed much better at identifying High Priority emails in terms of Recall rate, detecting a significantly higher number of urgent emails than the baseline model.

Figure (4.5): Comparison of High Priority Recall.



Analysis of Results

It can be seen that the DistilBERT model surpassed the Linear SVM model in performance on all metrics considered.

The largest improvement was noted in the recall of High Priority emails when the rate increased from 0.6900 to 0.8627. The importance of this improvement lies in the fact that the presence of urgent emails may be critical for making decisions or conducting business operations.

Additionally, the increase in Accuracy and Macro F1-score proves that the proposed model has achieved a balanced predictive performance across all priority classes.

4.9.4 Comparison of Model Architectures

Besides achieving higher predictive performance, the proposed model also provides architectural advantages over the traditional machine learning approaches.

Table (4.13): Comparison of the Baseline Models and the Proposed DistilBERT Model

Element	Traditional Baseline Models	Proposed DistilBERT Model
Number of Models	Two independent models	One unified model
Text Representation	TF-IDF	Contextual Embeddings
Feature Extraction	Manual	Automatic
Learning Strategy	Independent Learning	Multi-task Learning
Shared Representation	No	Yes
Feature Engineering Required	Yes	No
Context Understanding	Limited	High

Comparative Analysis

Unlike other traditional methods that need to have a distinct model for categorization and another one for prediction, the newly designed architecture of DistilBERT manages to do both at the same time, using a multi-task learning approach.

Furthermore, the new architecture of DistilBERT is able to learn its own context representation from email text data, without having any kind of manual input feature selection, thus offering a better understanding of the message's semantic meaning.

4.9.5 Overall Conclusion of the Comparison

The comparative analysis of the results reveals that the suggested DistilBERT multi-task model consistently outperformed the best traditional model for both classification tasks.

In terms of category classification, the suggested model has achieved a better result in terms of Macro F1-score and also considerably decreased Hamming Loss.

In terms of priority classification, the suggested model has improved such metrics as Accuracy, Macro F1-score, and, most importantly, High Priority Recall; thus, reducing the risk of missing urgent emails.

On the whole, the results shown in the tables (4.11 - 4.13) and graphics (4.10 – 4.13) clearly indicate the advantage of the suggested DistilBERT framework compared to the traditional Linear SVM model.

Overall, it is necessary to state that the experiments conducted proved the efficiency of the approach to combining contextual language representation with multi-task learning for intelligent email classification and priority prediction.

Section Analysis

Based on the comparative analysis, it is possible to state that the proposed DistilBERT multi-task model outperformed the best traditional baseline model in terms of classification tasks. The improvements are noticeable not only according to the quantitative metrics of the evaluation but also visually in terms of the graphics presented above that prove the superiority in terms of classification accuracy and a decrease in error rate.

Moreover, the proposed model combines category classification and priority prediction in one system without any need for manual feature engineering. These advantages make the proposed framework very useful in the field of intelligent email management systems.

Table (4.14): Summary of Final Experimental Results

Experiment	Main Result
Logistic Regression	Category Macro F1 = 0.8751
Linear SVM	Category Macro F1 = 0.9269
DistilBERT Multi-Task	Category Macro F1 = 0.9672
Priority Classification	Accuracy = 0.9529
Threshold Tuning	Best Threshold = 0.60
Explainable AI	LIME Successfully Applied
Prototype	Successfully Implemented

Table 4.14 contains an overview of major experiments performed within this chapter along with major findings from these experiments. The figure below represents the experimental process flow before proceeding to the results section discussion.

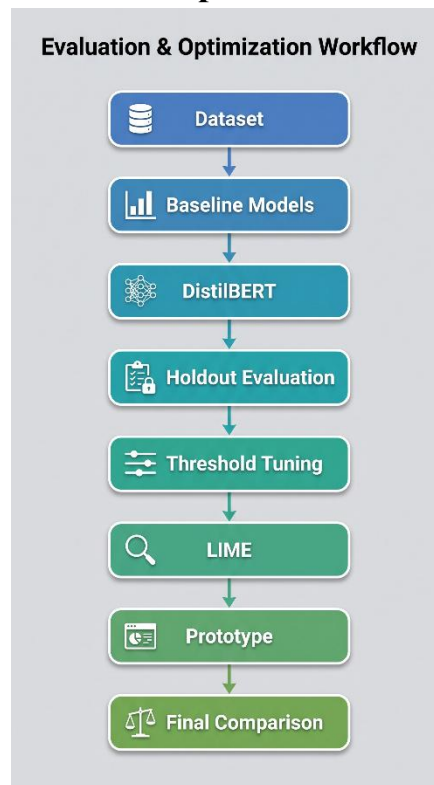
Figure (4.6): Overall Experimental Pipeline

Figure 4.5 presents the order of experiments performed after performing the data preparation, starting from the baseline models, followed by the suggested DistilBERT model, holdout validation, threshold tuning, explainability via LIME, prototyping, and model comparison.

4.10 Discussion

4.10.1 Introduction

In this research, the aim was to design an intelligent email management system that can classify emails into different categories based on multiple labels and priorities at once.

To test the performance of the suggested approach, it was compared with baseline approaches in terms of their accuracy through the same datasets and evaluation metrics.

The experimental results showed that the suggested DistilBERT multi-task model significantly outperformed the baselines. This section describes the reasons for the obtained results in relation to the applied model architecture and the entire experimental process.

4.10.2 Discussion of Category Classification Results

It is clear that the suggested model performed well in terms of the category classification task and outperformed the baseline models. The reason for this lies in the way contextualized language representations were generated by DistilBERT.

Contrary to TF-IDF techniques, which treat documents as separate word frequency vectors, DistilBERT considers semantic connections between words depending on the context they occur in. Therefore, it could differentiate between emails that had similar vocabulary but expressed different meaning or administrative intent.

Additionally, due to the use of contextual embeddings, the model managed to consider the coexistence of several topics in some emails, which is critical in a multi-label classification problem. This explains the increase in Macro F1-score in addition to the decrease in Hamming Loss.

4.10.3 Discussion of Prioritization Results

The proposed model was also shown to outperform the baseline Linear SVM model in the priority classification task.

One of the most practical results that can be highlighted is the improvement in the identification of High Priority emails. As High Priority emails demand urgent handling, the improvement in their identification enhances the efficiency of email intelligence solutions.

The contextual embeddings generated by DistilBERT allowed the model to identify the linguistic cues related to urgency in emails more effectively than statistical embeddings. The classification performance thus achieved was improved without compromising the performance on any other priority class.

4.10.4 The Impact of Multi-Task Learning

One significant contribution of this research is that a multitask learning approach was used.

In this case, instead of building two separate models, one shared DistilBERT encoder was used to represent the category classification task and the priority classification task, while the individual prediction heads were used for both tasks.

By using this architecture, the model learned some common semantic properties that could be useful for both tasks.

Apart from making the architecture simpler, it also ensured efficient learning without losing performance in any way.

4.10.5 Effect of Threshold Tuning

Threshold optimization was proven to be a good post-processing step after the training of the model.

The results obtained have indicated that the replacement of the default threshold with the optimal one increased the quality of category classification slightly but only at inference time, without retraining the model.

Though the effect observed was minor, the experiment proved the point that careful choice of the inference parameters can significantly increase the prediction quality almost without any additional costs.

4.10.6 Role of Explainable Artificial Intelligence

Model performance alone is not enough for numerous practical applications that need some clarity and trust from users.

That is why LIME was used to interpret each prediction made by the model after its training. It appeared that the model was using mostly words and phrases that were important for the understanding of the e-mail's content, not some random terms.

The model became more reliable thanks to the explanation of its behavior in a way understandable for a human.

In addition, the use of LIME helped analyze both true and false predictions.

4.10.7 Achievement of the Study's Objectives

As seen from the results of the experiment, the proposed methodology has accomplished all the stated goals that were formulated at the beginning of the research.

Namely, the following was done successfully:

- category classification and priority classification in one model;
- exceeded the performance of baseline machine learning models on the evaluation criteria;
- minimized classification errors for the task of category classification;
- increased the identification of High Priority emails;
- made the model more explainable with the help of Explainable Artificial Intelligence.

Thus, it can be concluded that the proposed methodology is an efficient tool for intelligent email management systems.

4.10.8 Summary of the Discussion

In general, these results show that the use of DistilBERT along with the multi-task learning approach gives some significant benefits over machine learning techniques.

These benefits were achieved due to the capabilities of the model to build context-based semantic embeddings, learn to perform several tasks at once, tune the

prediction decision by means of setting proper thresholds, and generate interpretable predictions via LIME.

As a result, all these features allowed the proposed model to reach a good trade-off between accuracy, computational cost, and interpretability.

4.11 Chapter Summary

In this chapter, the process of implementation and the experimental evaluation of the proposed intelligent email management system have been discussed.

Firstly, the implementation environment was introduced along with the data preprocessing steps and the design and training of the baseline machine learning models used as reference systems for comparison. Next, the proposed DistilBERT multi-task model design and implementation process were discussed in terms of the training process, validation approach, and model selection.

The experiments run on the independent Holdout Test Set have shown the superiority of the proposed model compared to the baseline models both for category classification and priority classification tasks. The proposed improvements were confirmed with the help of threshold tuning that improved category classification results at the inference step without any need for retraining.

The explainability of the model was addressed through Explainable AI based on LIME that helped interpret the predictions of the model and understand what textual features played a key role in the process.

As a result, the explanations provided showed that the proposed model used meaningful context-based information to classify the emails.

In conclusion, the results of the comparison of the baseline models with the proposed framework showed that the DistilBERT multi-task model is more effective at predicting while using one model to complete two tasks. Thus, the results prove that the goals of the research have been met and that there is a basis for creating an email management system.

Chapter 5: Conclusions, Recommendations, and Future Work

5.1 Introduction

This chapter is devoted to a summary of the main results of the conducted study. In addition, the scientific and practical contributions of the study are discussed, together with recommendations for improvement and ideas for further research.

5.2 Conclusions

The results from the experiments show that the proposed DistilBERT multi-task learning model managed to achieve the intended objectives for this study. The model managed to perform category classification and priority classification of emails in a single model design, thereby surpassing the traditional baseline models.

Comparing the results from the proposed model with those from Logistic Regression and Linear SVM, it was clear that the former had better classification results. In the case of category classification, the model was able to achieve a Macro F1 score of 0.9672 while the Hamming loss for the prediction was 0.0175, showing accuracy in multi-label predictions. In terms of priority classification, the model was able to achieve an accuracy of 0.9529, a Macro F1 score of 0.9327, and High Priority Recall of 0.8627.

The experiments also showed that the multi-task learning approach was effective since it was possible to use the same DistilBERT encoder to learn the contextual representation needed in classification tasks. This has made it easier to have a simpler architecture since the two models were not necessary.

Threshold tuning was another successful post-processing approach. The increase in the decision threshold for the categories from 0.50 to 0.60 resulted in a small increase in the Macro F1-score accompanied by a decrease in the Hamming Loss score without any additional training or changes in the architecture of the model.

Finally, the use of Explainable Artificial Intelligence (XAI) through the LIME method allowed for explaining the proposed model in terms of the text features used in each prediction.

Figure (5.1): Overall Framework of the Proposed Study

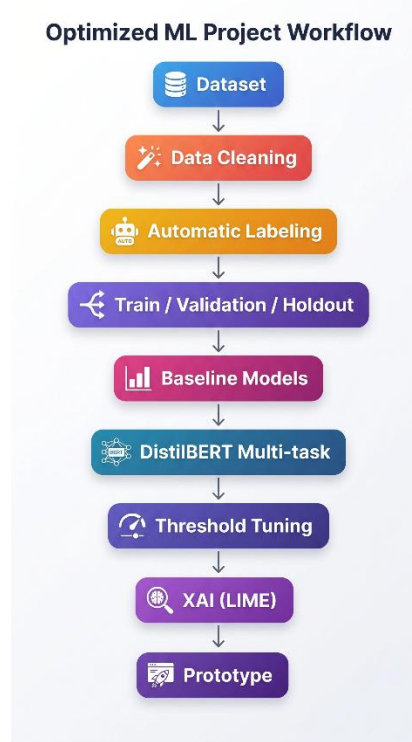


Figure 5.1 presents the workflow of the entire proposed study starting from the data preparation stage through model creation and optimization, its interpretation, and ending with prototype implementation.

5.3 Study Contributions

The research contributes to the scientific community in several ways that go beyond just improving classification results.

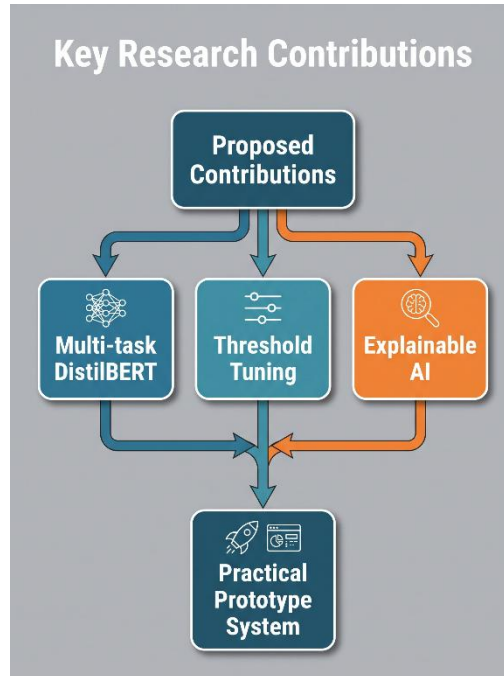
First of all, it develops an integrated multi-task learning system that can classify and predict the priority of emails in one step by using a single DistilBERT model.

Secondly, the research proves that contextual language models significantly outperform traditional machine learning techniques based on TF-IDF for intelligent email handling.

Thirdly, the research proves that the simplest possible inference-time tuning of a threshold improves multi-label classification performance with no additional costs at all.

Lastly, the research incorporates Explainable Artificial Intelligence (LIME) into the proposed system, making the individual predictions explainable.

Figure (5.2): Main Contributions of the Study



Summary of major scientific and practical achievements accomplished within the framework of the suggested approach, such as DistilBERT multi-task architecture, threshold tuning, Explainable Artificial Intelligence, and development of the practical prototype, are provided in Figure 5.2.

5.4 Recommendations

Based on the findings of this study, the following recommendations are proposed:

- Deploy the proposed DistilBERT model as an independent software component (such as a plug-in, add-on, or RESTful API) which can be plugged into various email management tools for automatic classification of incoming emails and their priorities.
- Perform threshold optimization as a regular post-processing operation when the decision thresholds are selected on the validation dataset to increase multi-label classification quality without re-training the model.

- Utilize Explainable Artificial Intelligence (XAI) approaches, such as LIME, to provide explanations of predictions in the form of words and phrases that impact the prediction most.
- Regularly retrain and validate the model with new email datasets to keep up with changes in communication patterns, vocabulary, and business needs without reducing prediction accuracy.
- Enhance the proposed approach by implementing other NLP techniques, including email summarization, automatic response suggestions, and task extraction.

5.5 Future Work

A number of promising research directions can be pursued to further develop the proposed framework.

- Consider applying more advanced transformer models, including BERT, RoBERTa, and DeBERTa, as well as lightweight language models, to evaluate the trade-off between predictive performance, computational cost, and inference time.
- Extend the proposed framework by supporting multilingual email classification, enabling the processing of emails in different languages using a unified model.
- Incorporate additional contextual features, such as email attachments, sender and recipient metadata, timestamps, conversation threads, and historical interactions, to improve category classification and priority prediction.
- Investigate adaptive threshold tuning techniques that take into account class distributions, prediction confidence, or user feedback to improve multi-label classification under evolving data distributions.
- Consider integrating Large Language Models (LLMs) into the proposed framework to support advanced email management tasks, such as email summarization, reply recommendation, information extraction, and task identification.
- Develop the proposed framework as a reusable software component (e.g., a plug-in, add-on, or RESTful API) and evaluate its performance in real-world email management applications.

5.6 Chapter Summary

In conclusion, this chapter presented the summary of findings of the study and proved that the designed DistilBERT multi-task model was able to accomplish the research objectives.

The findings showed great improvements compared to the machine learning algorithms in terms of classification and prioritization tasks. Moreover, tuning the threshold provided additional improvement in multi-label classification, while LIME provided increased interpretability of the model.

The research offers a useful approach towards intelligent email management with the use of context-aware language processing, multi-task learning, optimization during inference, and explainable artificial intelligence.

References

- Adnan, M., Imam, M. O., Javed, M. F., & Murtza, I. (2024). *Improving spam email classification accuracy using ensemble techniques: A stacking approach*. *International Journal of Information Security*, 23, 505–517. <https://doi.org/10.1007/s10207-023-00756-1>
- Ahmed, N., Amin, R., Aldabbas, H., Koundal, D., Alouffi, B., & Shah, T. (2022). *Machine learning techniques for spam detection in email and IoT platforms: Analysis and research challenges*. *Security and Communication Networks*, 2022, Article 1862333. <https://doi.org/10.1155/2022/1862333>
- Ali, N., Fatima, A., Shahzadi, H., Ullah, A., & Polat, K. (2021). *Feature extraction aligned email classification based on imperative sentence selection through deep learning*. *Journal of Artificial Intelligence and Systems*, 3, 93–114. <https://doi.org/10.33969/AIS.2021.31007>
- Bhalla, S., Kaur, M., & Sharma, V. (2021). *Automatic genre categorization of emails into predefined categories*. *Expert Systems with Applications*.
- Chuttur, M. Y., & Parianen, Y. (2022). *A comparison of machine learning models to prioritise emails using emotion analysis for customer service excellence*. *Knowledge Engineering and Data Science*, 5(1), Article 4. <https://doi.org/10.17977/um018v5i12022p41-52>
- Duong, T., Tran, A., Dang, V., Tran, T., Le, H.-S., & Nguyen, P. (2025). *AEEM: A deep learning approach to automated educational email management*. *Journal of Information Hiding and Multimedia Signal Processing*, 16(2).
- Fareed, B., Din, G. M., Khan, M. R. A., Fatima, T., & Shahid, S. (2025). *Machine learning-based email spam detection: Accuracy, overfitting and robustness analysis*. *European Journal of Applied Science, Engineering and Technology*, 3(6), 99–108. [https://doi.org/10.59324/ejaset.2025.3\(6\).06](https://doi.org/10.59324/ejaset.2025.3(6).06)
- Hirpara, D., & Polara, V. (2025). *InboxG: Smart mail classifier*. *International Journal on Science and Technology (IJSAT)*, 16(2).
- Iqbal, M., & Khan, A. (2022). *Email classification analysis using machine learning techniques*. *Data Science Journal*.
- Jain, U., Mishra, P., Dash, A., & Pandey, A. (2025). *Multi-label multi-class text classification-enhanced attention in transformers with knowledge distillation*. *Journal of Applied Research and Technology*, 23, 82–93.
- Jáñez-Martino, F., Alaiz-Rodríguez, R., González-Castro, V., Fidalgo, E., & Alegre, E. (2022). *A review of spam email detection: Analysis of spammer strategies and the dataset shift problem*. *Progress in Artificial Intelligence*. <https://doi.org/10.1007/s13748-022-00281-8>
- Kaddoura, S., Chandrasekaran, G., Popescu, D. E., & Duraisamy, J. H. (2022). *A systematic literature review on spam content detection and classification*. *PeerJ Computer Science*, 8, e830. <https://doi.org/10.7717/peerj-cs.830>

Khare, N., Singh, S., Mishra, R., Prakash, S., & Dixit, P. (2022). *E-mail assistant: Automation using robotic process automation (RPA)*. In *2022 International Conference on Decision Aid Sciences and Applications (DASA)* (pp. 511–516). IEEE. <https://doi.org/10.1109/DASA54658.2022.9765017>

Salloum, S., Gaber, T., Vadera, S., & Shaalan, K. (2022). *A systematic literature review on phishing email detection using natural language processing techniques*. *IEEE Access*, 10, 36433–36460. <https://doi.org/10.1109/ACCESS.2022.3165193>

Singh, M., Cambronero, J., Gulwani, S., Le, V., & Verbruggen, G. (2023). *EmFore: Online learning of email folder classification rules*. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management (CIKM '23)* (pp. 4262–4273). ACM. <https://doi.org/10.1145/3583780.3614863>

Wang, L., Zhang, H., Zhang, W., Dai, L., Liang, Y., & Shi, H. (2024). *Deep active learning for multi-label text classification*. *Neural Computing and Applications*.