International Journal of Recent Development in Engineering and Technology
Website: www.ijrdet.com (ISSN 2347-6435 (Online) Volume 15, Issue 08, August 2026)

An NLP-Based Intelligent System for Automated Medical Report Classification and Clinical Information Extraction Using TF-IDF and Logistic Regression

Bhargava P M[1], Basavanna M[2], Poornima B H[3], Bindu K B[4], T Chandraiah[5], Latharani T R[6]

[1]*Department of Studies in Computer Applications (MCA), Davangere University, Davangere, Karnataka, India.*
[2,3,4]*Department of Studies in Computer Science, Davangere University, Davangere, Karnataka, India.*
[5]*Department of Computer science, Yuvaraja College, University of Mysore, Mysore, Karnataka, India.*
[6]*Department of Computer Science and Engineering, Ramiah University of Applied Sciences, Bangalore, Karnataka, India.*

***Abstract*— A Medical Report Analyzer Using Natural Language Processing is developed to make medical reports easier to understand and analyze. The system accepts an uploaded medical report, identifies its category from 14 different medical categories, and extracts useful information such as diseases, medicines, and laboratory test values. It also creates a simple summary of the report and provides dietary suggestions based on the identified condition. The classification model uses TF-IDF with Logistic Regression and achieved around 99.5% accuracy during 10-fold cross-validation. The complete system was tested for its functionality, security, and performance. The main advantage of the proposed system is that it brings classification, information extraction, summarization, and dietary recommendations together in a single application.**

***Keywords*—Clinical Text Classification, Information Extraction, Logistic Regression, Medical Report Analysis, Natural Language Processing, Text Summarization, TF-IDF.**

I. INTRODUCTION

The quick shift to digital in healthcare has led to a huge amount of electronic medical reports, such as discharge summaries, laboratory results, and prescriptions, which are produced faster than they can realistically be reviewed by hand. Patients and even busy clinicians often struggle to quickly understand what a report says, what condition it points to, and what to do about it, because most of this information exists as unstructured, free-form text rather than searchable, structured data.

By using advanced technologies, Natural Language Processing (NLP) and Machine Learning (ML) have emerged as effective tools for automating medical report analysis. By reading the text of an uploaded report and analyzing it through trained classification and extraction algorithms, it becomes possible to identify the medical category, disease, medicines, and laboratory findings quickly and accurately.

This paper focuses on building an end-to-end system capable of classifying a medical report into one of fourteen medical categories, extracting key clinical details, and explaining the findings in plain language.

Traditional manual review of medical reports takes considerable time and effort and is not always consistent, particularly for patients without a clinical background. With NLP and ML, reports can be analyzed and explained quickly, helping patients and clinicians take informed action sooner.

The proposed Medical Report Analyzer utilizes AI and NLP techniques to automate the analysis of medical reports through a user-friendly web application developed using Flask and Bootstrap. The system allows users to upload PDF medical reports, extracts and preprocesses the text, identifies diseases, medications, and laboratory values, classifies reports into appropriate medical categories, generates AI-based summaries, and maintains prediction history for future reference. Additionally, users can download a structured PDF containing the extracted information and analysis results. By combining advanced NLP models with an intuitive web interface, the proposed system aims to improve the efficiency, accuracy, and accessibility of medical report interpretation while reducing the manual effort required by healthcare professionals and providing valuable insights for patients and medical practitioners.

The rest of the paper is organized as follows: Section II presents the related work and reviews existing methods for pest detection in agriculture. Section III presents the methodology for the proposed research. Section IV describes the Technology Used, while Section V discusses the Results. Section VI presents the conclusion of the paper. Finally, Section VII includes References.

A. Problem Statement

Healthcare generates large volumes of unstructured medical reports, including laboratory results, diagnostic reports, discharge summaries, and clinical notes.

International Journal of Recent Development in Engineering and Technology
Website: www.ijrdet.com (ISSN 2347-6435 (Online) Volume 15, Issue 08, August 2026)

Manually reviewing these documents to identify diseases, medications, and laboratory findings is time-consuming and error-prone. Although Electronic Health Record (EHR) systems store such documents efficiently, they generally lack the intelligence to automatically extract or interpret the clinical information within them.

B. Objectives

1. To develop an intelligent Medical Report Analyzer using Natural Language Processing (NLP) for automated analysis of clinical reports.
2. To extract key clinical information such as diseases, medications, symptoms, and laboratory values from uploaded medical reports using NLP techniques.
3. To generate AI-based summaries and classify medical reports into appropriate disease categories for quick and accurate interpretation.
4. To design and implement a secure web application using Flask and Bootstrap that enables users to upload, analyze, and download structured medical report analyses.

II. Related Work

1. Almazaydeh, Abuhelaleh, Al Tawil & Elleithy, "Clinical Text Classification with Word Representation Features and Machine Learning Algorithms," 2023. The authors compared SVM, Naive Bayes, Logistic Regression, and k-Nearest Neighbors across three text representations (Bag-of-Words, TF-IDF, Word2Vec) on medical transcription data, finding Word2Vec + k-NN strongest overall, though TF-IDF-based linear models stayed competitive at a fraction of the training cost
2. Eshza, "Multiclass-text-classification" (GitHub repository), this open-source project trained four models (Naive Bayes, SVM, Random Forest, Logistic Regression) to sort clinical notes into five medical specialties, and specifically observed Naive Bayes and Random Forest collapsing predictions toward the majority class wherever specialty vocabulary overlapped. This is the same overlap failure mode later verified independently in this project's own testing between closely related categories.
3. Kalra, Li & Tizhoosh, "Automatic Classification of Pathology Reports using TF-IDF Features," 2019, the authors built a 1,949-report pathology dataset spanning 37 diagnosis categories and classified it using TF-IDF features with SVM, XGBoost, and Logistic Regression, reaching 92% accuracy with XGBoost and 87% with linear SVM.
4. "Large Language Models for Healthcare Text Classification: A Systematic Review," 2025, this review surveys how LLMs are being applied across healthcare text classification tasks, providing broader context for where classical ML sits relative to newer transformer-based approaches. It supports this project's positioning of TF-IDF + Logistic Regression as a deliberately lighter-weight alternative to LLM-based classification.
5. Sarim, "Predicting Diseases with TF-IDF and One-Hot Encodings Using Machine Learning," 2025. This applied walkthrough demonstrates disease prediction from symptom text using TF-IDF alongside one-hot encoding, reinforcing that TF-IDF remains a standard, accessible starting point for symptom/disease-text classification tasks similar to this project's core prediction engine.
6. "Comparative Study of Machine Learning Models for Textual Medical Note Classification," Computers journal, 2026. This study benchmarked Random Forest, Logistic Regression, Multinomial Naive Bayes, and SVM on 9,633 labeled medical notes across four disease categories, finding Logistic Regression strongest overall (84.7% accuracy) while specifically flagging lexical overlap between digestive and neurological notes as a source of confusion.
7. "Enhancing Clinical Named Entity Recognition via Fine-Tuned BERT and Dictionary-Infused Retrieval-Augmented Generation," Electronics (MDPI), 2025. This work combines fine-tuned BERT with a dictionary-based retrieval layer to improve clinical entity recognition accuracy, illustrating one direction MedInsight AI's rule-based extraction modules could evolve toward if a future iteration adopted deep learning.
8. Yu, Hu, Lu, Sun & Yuan, "BioBERT Based Named Entity Recognition in Electronic Medical Record," 2019, the authors applied a BioBERT + BiLSTM-CRF pipeline to the I2B2 2010 clinical entity tagging challenge, reaching an F1 score of 87.1% for identifying problems, treatments, and tests in medical records. This quantifies the accuracy ceiling available from transformer-based extraction, against which this project's simpler regex-based extraction is a deliberate, resource-conscious trade-off.

III. Methodology

The paper's methodology involves a web-based application designed to make medical report analysis faster and easier.

International Journal of Recent Development in Engineering and Technology
Website: www.ijrdet.com (ISSN 2347-6435 (Online) Volume 15, Issue 08, August 2026)

It uses Natural Language Processing (NLP) and Artificial Intelligence (AI) to analyze uploaded PDF medical reports and extract useful information. Users can securely register, log in, and upload their reports through a simple interface. The system processes the extracted text by cleaning it, tokenizing it, and removing unnecessary words. It then identifies important medical information such as diseases, medicines, symptoms, and laboratory findings. In addition, the system summarizes the report, predicts the relevant disease category, and keeps a record of previous predictions. Users can also download the analyzed information as a structured PDF report for easy reference.

1. Requirement Analysis

The first phase focuses on identifying the functional and non-functional requirements of the Medical Report Analyzer. The system is required to allow users to upload medical reports, classify them into appropriate categories, extract important medical information, generate summaries, provide health advice, and securely store the results.

2. System Design

In this phase, the overall architecture and workflow of the system are designed. The design includes the user interface, report-upload module, text preprocessing module, TF-IDF feature extraction, classification module, medical entity extraction, summarization, health advice, database, and result management. The flow of data between these modules is also defined.

3. Implementation (Coding)

The system is implemented using suitable programming languages, libraries, and frameworks. Publicly available medical text data is collected and preprocessed using lowercasing, tokenization, stopword removal, and lemmatization. TF-IDF is then used to convert the processed text into numerical features. Different machine learning algorithms are trained and compared to select the most suitable classification model.

4. Integration and Testing

After individual modules are developed, they are integrated into a complete system. Regular expressions are used to extract diseases, medicines, and laboratory values from medical reports. The integrated system is tested to verify report uploading, text processing, classification, entity extraction, summarization, health advice, and data storage. Functional and performance testing is carried out to ensure reliable results.

5. Deployment

Once testing is completed successfully, the Medical Report Analyzer is deployed as a web-based application. Users can register and log in, upload their medical reports, and obtain the analysis through the application. The system provides the predicted category, extracted medical information, summary, and relevant health advice.

6. Maintenance

After deployment, the system is maintained to ensure continuous and reliable operation. Errors identified during use are corrected, the classification model can be updated with new medical data, and system features can be improved when required. Security and performance are also monitored to keep the application effective over time.

The system architecture shown in Fig. 2 follows a pipeline in which users first authenticate and upload medical reports through the web application. The uploaded reports are processed through the information extraction module to obtain relevant medical text. The extracted text is then preprocessed and analyzed to detect diseases and identify important medical information.

The proposed methodology uses NLP techniques and machine learning to classify medical reports and extract medicines and laboratory values. The analyzed information is then passed to the result summarization module to generate a clear and understandable summary. Finally, the summarized results are displayed to the user through the web application, helping them understand the important findings of the medical report more easily.

A. Architecture Diagram

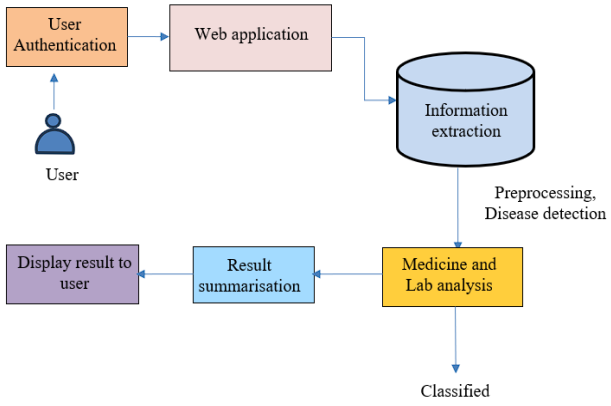

Fig.1 System Architecture

International Journal of Recent Development in Engineering and Technology
Website: www.ijrdet.com (ISSN 2347-6435 (Online) Volume 15, Issue 08, August 2026)

1) Workflow Explanation

The system begins with user authentication, allowing authorized users to securely register and log in before accessing the application. After login, users can upload medical reports in PDF format, from which the system extracts text using PDF processing techniques.

The extracted text is cleaned through preprocessing, including tokenization, stop-word removal, and lemmatization. NLP techniques are then used to identify diseases, medicines, symptoms, and laboratory values, while machine learning methods classify the report into a suitable disease category.

The system generates an AI-based summary highlighting the important findings of the medical report. Finally, the extracted information, summary, and classification results are stored in the database and provided as a structured PDF report for future reference.

2) Algorithm:

Input: Medical report (PDF file or pasted text), authenticated user session.

Output: Predicted category, detected diseases, medicines, lab values, plain-language summary, diet advice, downloadable PDF

Step1: Start the medical report analysis process.

Step2: Upload a medical report PDF or paste the report text into the system.

Step3: Extract the report text and ensure it is not empty. Clean and preprocess the text by removing unnecessary content and getting it ready for analysis.

Step4: Convert the processed text into TF-IDF features for machine learning.

Step5: Classify the report using the trained Logistic Regression model and calculate the confidence score.

Step6: Assign the final report category based on the prediction confidence.

Step7: Extract important medical information such as diseases, medicines, and lab values.

Step8: Generate a clear and easy-to-understand summary of the medical report.

Step9: Provide diet and lifestyle recommendations based on the detected medical condition.

Step10: Save the analysis results and create a downloadable PDF report. Show the final report to the user and end the process.

IV. Technology Used

The project is implemented using widely adopted open-source technologies that support Natural Language Processing, web development, and document processing.

1) Python

Python is the primary programming language used to develop the application. It provides extensive libraries for Natural Language Processing, machine learning, and web development.

2) Flask

Flask is a lightweight Python web framework used to develop the backend of the application. It manages user requests, processes medical reports, and communicates with the database.

3) Bootstrap 5

Bootstrap 5 is used to create a responsive and user-friendly web interface. It ensures that the application works efficiently across different devices and screen sizes.

Libraries Used

1) spaCy

spaCy is an advanced Natural Language Processing library used for text preprocessing, tokenization, and Named Entity Recognition (NER) to extract medical entities.

2) NLTK

The Natural Language Toolkit (NLTK) is used for preprocessing medical text, including tokenization, stop-word removal, and lemmatization.

3) PyMuPDF / pdfplumber

PyMuPDF and pdfplumber are used to extract textual content from uploaded PDF medical reports for further NLP analysis.

4) Pandas

Pandas is used for data manipulation, processing, and organizing extracted medical information into structured formats.

5) NumPy

NumPy provides efficient numerical computation and supports mathematical operations required during data processing.

International Journal of Recent Development in Engineering and Technology
Website: www.ijrdet.com (ISSN 2347-6435 (Online) Volume 15, Issue 08, August 2026)

V. RESULTS

The proposed Medical Report Analyzer successfully extracted meaningful clinical information from uploaded medical reports. The application demonstrated accurate identification of diseases, prescribed medicines, laboratory findings, and important medical observations. The AI-generated summaries effectively reduced lengthy reports into concise and informative descriptions while preserving clinically relevant information.

The disease classification module categorised reports into appropriate disease groups based on the extracted entities and contextual understanding of the clinical text. Prediction history was stored successfully, allowing users to review previous analyses.

The web interface developed using Flask and Bootstrap provided smooth navigation, responsive design, and easy interaction for users with minimal technical knowledge.

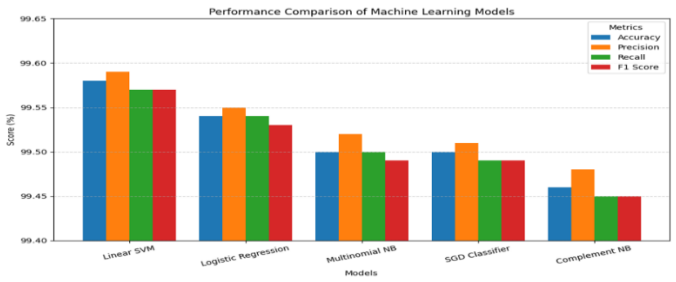

Fig.2 Model Accuracy Comparison

Table I Comparison Of Classification Models Based On Accuracy, Precision, Recall, And F1-Score

Model	Accuracy	Precision	Recall	F1 Score
Linear SVM	99.58	99.59	99.57	99.57
Logistic Regression	99.54	99.55	99.54	99.53
Naive Bayes (Multinomial)	99.50	99.52	99.50	99.49
SGD Classifier	99.50	99.51	99.49	99.49
Naive Bayes (Complement)	99.46	99.48	99.45	99.45

1) Snapshots:

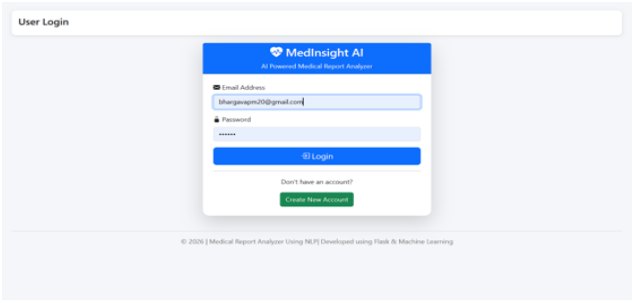

Fig 3. Home Page

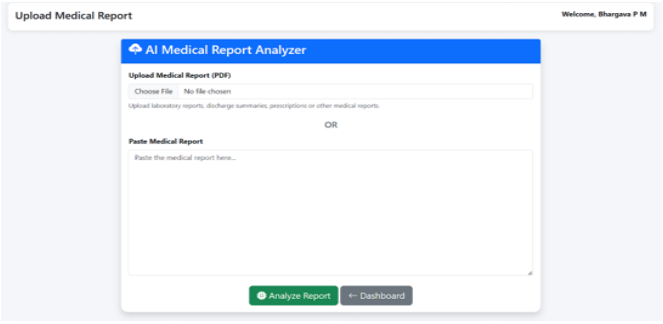

Fig 4. Upload Page

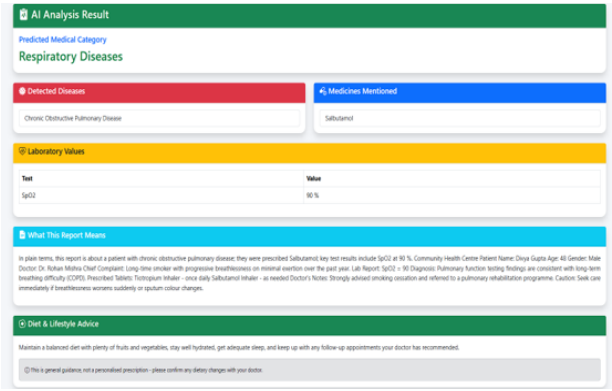

Fig 5. Result Page

VI. CONCLUSION

The rapid growth of Electronic Health Records (EHRs) and digital healthcare systems has increased the need for intelligent solutions capable of efficiently analysing medical reports.

The proposed Medical Report Analyser addresses this need by leveraging Artificial Intelligence (AI) and Natural Language Processing (NLP) to automatically extract diseases, medications, laboratory findings, and other clinical information from unstructured medical reports. The system also generates concise AI-based summaries and classifies medical reports, enabling healthcare professionals and patients to quickly understand important clinical information while reducing manual effort and improving efficiency.

REFERENCES

[1] L. Almazaydeh, M. Abuhelaleh, A. Al Tawil, and K. Elleithy, "Clinical Text Classification with Word Representation Features and Machine Learning Algorithms," *International Journal of Online and Biomedical Engineering (iJOE)*, vol. 19, no. 4, pp. 65–76, 2023. Available: https://online-journals.org/index.php/i-joe/article/view/36099

[2] eshza, "MultiClass-text-classification," GitHub repository, 2023. Available: https://github.com/eshza/MultiClass-text-classification

[3] S. Kalra, L. Li, and H. R. Tizhoosh, "Automatic Classification of Pathology Reports using TF-IDF Features,"*arXiv preprint*,arXiv:1903.07406, 2019. Available: https://arxiv.org/abs/1903.07406

[4] "Large Language Models for Healthcare Text Classification: A Systematic Review," *arXiv preprint*, arXiv:2503.01159, 2025. Available: https://arxiv.org/abs/2503.01159

[5] M. Sarim, "Predicting Diseases with TF-IDF and One-Hot Encodings Using Machine Learning, "*Medium*, 2025. Available:https://medium.com/@Muhammad.Sarim/predicting-diseases-with-tf-idf-and-one-hot-encodings-using-machine-learning-8991b371bcab

[6] "Comparative Study of Machine Learning Models for Textual Medical Note Classification," *Computers*, vol. 15, no. 1, art. 7, 2026. Available: https://doi.org/10.3390/computers15010007

[7] "Enhancing Clinical Named Entity Recognition via Fine-Tuned BERT and Dictionary-Infused Retrieval-Augmented Generation," *Electronics*, vol. 14, no. 18, art. 3676, 2025. Available: https://www.mdpi.com/2079-9292/14/18/3676

[8] X. Yu, W. Hu, S. Lu, X. Sun, and Z. Yuan, "BioBERT Based Named Entity Recognition in Electronic Medical Record," in *Proc. 2019 10th Int. Conf. on Information Technology in Medicine and Education (ITME)*, Qingdao, China, 2019, pp. 49–52. Available: https://ieeexplore.ieee.org/document/8965108/