TRANSLATING MEDICAL NOTES INTO ICD-10 CODES

PHASE II REPORT

Submitted by

SHUBALEKHA G	**(2116220701275)**
SREENIDHI K	**(2116220701285)**

in partial fulfillment for the award of the degree

of

BACHELOR OF ENGINEERING

in

COMPUTER SCIENCE AND ENGINEERING

RAJALAKSHMI ENGINEERING COLLEGE

ANNA UNIVERSITY, CHENNAI

MAY 2025

RAJALAKSHMI ENGINEERING COLLEGE, CHENNAI

BONAFIDE CERTIFICATE

Certified that this Project titled **“Translating Medical Notes into ICD-10 Codes”** is the bonafide work of **“ SHUBALEKHA G (2116220701275),SREENIDHI K (2116220701285)”** who carried out the work under my supervision. Certified further that to the best of my knowledge the work reported herein does not form part of any other thesis or dissertation on the basis of which a degree or award was conferred on an earlier occasion on this or any other candidate.

SIGNATURE
Dr. P. Kumar., M.E., Ph.D.,
HEAD OF THE DEPARTMENT
Professor
Department of Computer Science
and Engineering,
Rajalakshmi Engineering College,
Chennai - 602 105.

SIGNATURE
Dr. M. Rakesh Kumar., M.E., Ph.D.,
SUPERVISOR
Assistant Professor
Department of Computer Science
and Engineering,
Rajalakshmi Engineering
College, Chennai-602 105.

Submitted to Project Viva-Voce Examination held on ___________________

Internal Examiner **External Examiner**

ABSTRACT

"**Translating Medical Notes into ICD-10 Codes**" is an intelligent system designed to automate the diagnosis coding process by transforming unstructured medical text into structured ICD-10 codes. Leveraging advanced Natural Language Processing (NLP) and Machine Learning (ML) techniques, this system analyzes clinical notes and predicts relevant diagnosis codes, significantly improving accuracy and efficiency in medical documentation. The platform integrates BERT (Bidirectional Encoder Representations from Transformers) to extract contextual meaning from clinical language and applies multi-label classification models such as Logistic Regression, Random Forest, and Support Vector Machines to handle cases where a single medical note may correspond to multiple diagnoses. Ensemble learning methods are utilized to enhance performance and robustness. The system's performance is evaluated using precision, recall, and F1-score metrics to ensure high reliability in code assignment. By automating a critical aspect of healthcare data processing, this project addresses inefficiencies in manual coding, reduces human error, and supports scalable deployment across clinical systems — ultimately contributing to improved patient care and administrative efficiency.

ACKNOWLEDGMENT

Initially we thank the Almighty for being with us through every walk of our life and showering his blessings through the endeavor to put forth this report. Our sincere thanks to our Chairman **Mr. S. MEGANATHAN, B.E, F.I.E.**, our Vice Chairman **Mr. ABHAY SHANKAR MEGANATHAN, B.E., M.S.,** and our respected Chairperson **Dr. (Mrs.) THANGAM MEGANATHAN**, **Ph.D.,** for providing us with the requisite infrastructure and sincere endeavoring in educating us in their premier institution.

Our sincere thanks to **Dr. S.N. MURUGESAN, M.E., Ph.D.,** our beloved Principal for his kind support and facilities provided to complete our work in time. We express our sincere thanks to **Dr. P. KUMAR, M.E., Ph.D.**, Professor and Head of the Department of Computer Science and Engineering for his guidance and encouragement throughout the project work. We convey our sincere and deepest gratitude to our internal guide, **Dr. S. VINOD KUMAR , M.Tech., Ph.D.,** Professor of the Department of Computer Science and Engineering. Rajalakshmi Engineering College for his valuable guidance throughout the course of the project. We are very glad to thank our Project Coordinator, **Mr. M. RAKESH KUMAR** Assistant Professor Department of Computer Science and Engineering for his useful tips during our review to build our project.

SHUBALEKHA G	**2116220701275**
SREENIDHI K	**2116220701285**

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

S. No	ABBR	Expansion
1	AI	Artificial Intelligence
2`	API	Application Programming Interface
3	ICD	International Classification of Diseases
4	NLP	Natural Language Processing
5	TF-IDF	Term Frequency - Inverse Document Frequency
6	UI	User Interface
7	LR	Logistic Regression
8	DFD	Data Flow Diagram
9	ML	Machine Learning
10	RF	Random Forest
11	SQL	Structure Query Language
12	SVM	Support Vector Machine

CHAPTER 1

INTRODUCTION

1.1 GENERAL

" Translating Medical Notes into ICD-10 Codes " is an innovative solution designed to streamline the process of assigning standardized ICD (International Classification of Diseases) codes to clinical symptom descriptions. This system addresses the challenges faced by healthcare professionals in accurately mapping patient-reported symptoms to appropriate diagnostic codes, which is essential for healthcare documentation, billing, and epidemiological research.By leveraging advanced machine learning techniques such as **Logistic Regression** and **Random Forest Classifier**, the platform enables efficient and accurate prediction of ICD codes based on textual descriptions of patient symptoms. The system incorporates **Natural Language Processing (NLP)** methods including **TF-IDF vectorization**, **stopword removal**, and **lemmatization** to process and analyze unstructured clinical text data effectively.The core functionality involves training a classifier on a carefully curated and labeled dataset of symptom–ICD code pairs. The system is further enhanced with a **web-based user interface** developed using **Flask framework**, which allows users to input symptom descriptions and receive real-time predicted ICD codes. Extensive evaluation metrics such as **accuracy**, **precision**, **recall** and **confusion matrix** ensure the reliability and robustness of the model.By automating the ICD code assignment process, the **ICD Code Prediction System** reduces manual workload, improves coding accuracy, enhances healthcare documentation, and contributes to more efficient clinical workflows, ultimately supporting better healthcare delivery and patient management.

1.2 OBJECTIVE

The objective of the " Translating Medical Notes into ICD-10 Codes " is to develop a robust machine learning-based application that accurately predicts ICD diagnosis codes from clinical symptom descriptions. The system aims to:

- Automate and simplify the ICD coding process for healthcare professionals
- Enhance the accuracy and efficiency of disease coding
- Minimize manual errors and reduce the workload on clinical coders
- Provide a user-friendly web interface for seamless interaction
- Ensure model reliability through extensive performance evaluation
- Support healthcare documentation, epidemiological studies, and healthcare billing with accurate and standardized diagnostic codes

By leveraging NLP techniques and machine learning algorithms such as Logistic Regression and Random Forest, the system ensures high prediction accuracy, improves clinical workflows, and contributes to more structured and efficient healthcare data management.

1.3 EXISTING SYSTEM

Currently, the assignment of ICD codes is predominantly a manual process performed by trained medical coders. These methods heavily rely on human expertise and interpretation of medical records, which often leads to:

- Time-consuming and labor-intensive workflows
- Inconsistencies and inaccuracies due to subjective interpretation
- Limited scalability in large healthcare settings

- Risk of human errors and misclassification
- Increased workload for healthcare providers and administrative staff

Additionally, existing **rule-based automated coding systems** lack flexibility and fail to accurately interpret complex, unstructured symptom descriptions. These systems are not adaptive to evolving clinical language and often require constant manual updates. Consequently, there is a growing need for an **intelligent, accurate, and automated solution** that can process free-text symptom descriptions, predict appropriate ICD codes efficiently, and reduce the burden on healthcare professionals — thereby improving overall healthcare documentation quality and operational efficiency.

CHAPTER 2
LITERATURE SURVEY

Automatic Clinical Coding: A Systematic Literature Review [1] (2020) by Jimeno-Yepes. This paper presents a comprehensive review of various automatic clinical coding systems that map clinical narratives to ICD codes. It highlights multiple machine learning and natural language processing techniques, including Support Vector Machines (SVM), Conditional Random Fields (CRF), and deep learning models. The study emphasizes the importance of feature engineering, data quality, and domain-specific language understanding in achieving accurate ICD code predictions. However, it notes that limitations include the need for large, labeled datasets and challenges in handling ambiguous clinical language, which can affect prediction accuracy.

Deep Learning Models for Automated ICD Coding of Clinical Texts [2] (2021) by Mullenbach,J.,etal. This study introduces the use of Convolutional Neural Networks (CNN) and Attention mechanisms for automatic ICD coding. The proposed model captures contextual relationships within clinical text, improving the prediction of ICD codes from discharge summaries. The paper demonstrates how deep learning models outperform traditional methods in complex, multi-label classification tasks. A key limitation identified is the computational complexity and high resource requirements for training deep models on large datasets, which can limit their scalability in smaller clinical settings.

An Ensemble Machine Learning Approach for ICD Code Assignment [3] (2022) by Zhang,Y.,etal. This paper explores the integration of ensemble learning techniques, such as Random Forest and Gradient Boosting, for ICD code assignment. The ensemble approach improves prediction accuracy by combining multiple classifiers and reducing individual model bias. The study highlights the effectiveness of ensemble methods in dealing with diverse and imbalanced clinical datasets. However, it also acknowledges increased training time and model complexity as trade-offs, requiring careful tuning of hyperparameters and computational resources.

Natural Language Processing for Healthcare: A Review [4] (2020) by Wang, Y., et al. The paper reviews various applications of NLP in healthcare, including clinical documentation, information extraction, and disease coding. It emphasizes the use of TF-IDF, word embeddings, and named entity recognition (NER) techniques for extracting meaningful insights from clinical narratives. The study highlights the potential of NLP methods in improving efficiency and accuracy in healthcare documentation. However, challenges such as handling domain-specific language, data privacy concerns, and the requirement of large annotated datasets are discussed as key limitations.

Automated Disease Coding in Clinical Narratives Using Machine Learning [5] (2021) byJo,Y.,etal. This study applies Support Vector Machines (SVM) and Logistic Regression classifiers for automated ICD coding. The authors demonstrate that machine learning models, when combined with effective text preprocessing methods such as stopword removal and lemmatization, can accurately predict ICD codes from clinical notes. The paper emphasizes the importance of high-quality feature extraction and labeled data in achieving reliable performance. A limitation noted is that traditional ML models may struggle with long or complex narratives without advanced contextual understanding.

Clinical Text Classification Using TF-IDF and Random Forest [6] (2020) by Alharbi, A.,etal. The paper investigates the use of TF-IDF vectorization combined with Random Forest classifiers to categorize clinical text into ICD codes. Results show that this approach performs well on moderately sized datasets and offers interpretability in feature importance analysis. However, the study acknowledges that the model's performance drops when handling highly imbalanced or noisy datasets, indicating the need for dataset balancing and noise reduction strategies.

Automated ICD Coding Using Pre-trained Language Models [7] (2022) by Si, Y., et al. This paper evaluates the use of BERT-based models (Bidirectional Encoder Representations from Transformers) for ICD code prediction. The authors show that transfer learning from large biomedical text corpora significantly boosts coding accuracy and model robustness. The study demonstrates the advantage of contextual embeddings in understanding clinical language. However, the high resource requirements for fine-tuning and model deployment pose challenges for smaller institutions with limited computational capacity.

A Review on Machine Learning Approaches for Medical Text Classification [8] (2021) byZhang,X.etal. The paper provides an overview of supervised, unsupervised, and deep learning methods applied to medical text classification. Techniques such as Naïve Bayes, Logistic Regression, and Decision Trees are discussed, along with their strengths and limitations. The review emphasizes the importance of choosing appropriate models based on data characteristics and task complexity. Limitations highlighted include feature sparsity, class imbalance, and the difficulty in interpreting complex deep models.

Improving Clinical Text Classification with Hybrid Models [9] (2023) by Chen, M., et al.This study proposes a hybrid approach that combines TF-IDF with deep learning models to enhance ICD code prediction. By integrating traditional feature-based methods with neural networks, the model improves performance while maintaining computational efficiency. The authors highlight that hybrid models can balance interpretability and accuracy. However, they also point out that integrating different model architectures requires careful design and extensive validation to ensure reliability.

Automatic Coding of Clinical Texts: Current Challenges and Future Directions [10] (2020)byAfzal,N.,etal. This paper outlines key challenges in automatic ICD coding, including data quality, language ambiguity, and model generalizability across institutions. It discusses current solutions such as ensemble models, transfer learning, and domain adaptation techniques. The authors emphasize the need for collaborative datasets, standardization of annotations, and explainable AI models to enhance trust and adoption in clinical practice. The paper concludes that addressing these challenges is crucial for deploying reliable and scalable ICD coding systems in healthcare.

CHAPTER 3

PROPOSED SYSTEM

3.1 GENERAL

Translating Medical Notes into ICD-10 Codes is an innovative solution designed to automate and streamline the assignment of standardized ICD codes based on clinical symptom descriptions. It leverages advanced machine learning algorithms such as **Logistic Regression** and **Random Forest Classifier** to predict accurate diagnostic codes from unstructured clinical text with high precision.

The system utilizes **Natural Language Processing (NLP)** techniques, including **TF-IDF vectorization**, **stopword removal**, and **lemmatization**, to effectively process and analyze symptom descriptions provided by healthcare professionals. Through these preprocessing methods, the model extracts key features from the text and maps them to the most relevant ICD codes.

A carefully curated and labeled dataset containing symptom–ICD code pairs is used to train and evaluate the classification models. To ensure user accessibility and practicality, the system is integrated into a **Flask-based web application**, providing an intuitive interface where users can input patient symptoms and obtain predicted ICD codes in real time.

By improving accuracy and efficiency in ICD code assignment, this system reduces manual workload, minimizes coding errors, enhances healthcare documentation, and supports better healthcare data management. Ultimately, the **ICD Code Prediction System** fosters more reliable and streamlined clinical workflows, contributing to improved healthcare delivery.

3.2 SYSTEM ARCHITECTURE DIAGRAM

The system architecture **(Fig 3.1)** for the **ICD Code Prediction System using Machine Learning** integrates machine learning techniques to ensure accurate and efficient prediction of ICD codes from clinical symptom descriptions. The architecture consists of several key phases involving user interaction, data processing, machine learning model training, and result delivery.It begins with **data collection and labeling**, where a curated dataset containing symptom descriptions paired with corresponding ICD codes is prepared. The next phase is **data preprocessing**, which involves cleaning the text data, handling inconsistencies, removing stopwords, and performing lemmatization to normalize the input text. Following preprocessing, **feature extraction** is performed using **TF-IDF vectorization**, converting the processed text into numerical feature vectors suitable for machine learning models. The system then proceeds with **classification and model training** using algorithms such as **Logistic Regression** and **Random Forest** (chosen for their high accuracy and interpretability in text classification tasks). The performance of the models is evaluated using **accuracy metrics**, **confusion matrix**, and **classification reports** to ensure robustness and reliability. Once optimized, the final trained model is deployed through a **Flask-based**

web application, enabling real-time ICD code prediction based on user-provided symptom descriptions.

All relevant data, including the trained models, predictions, and performance evaluations, are stored in a **centralized database**. The web server facilitates secure interaction between the user interface and the backend model. The system's features include reliable dataset labeling, efficient text preprocessing, robust machine learning classification, and a user-friendly web interface for ICD code prediction.

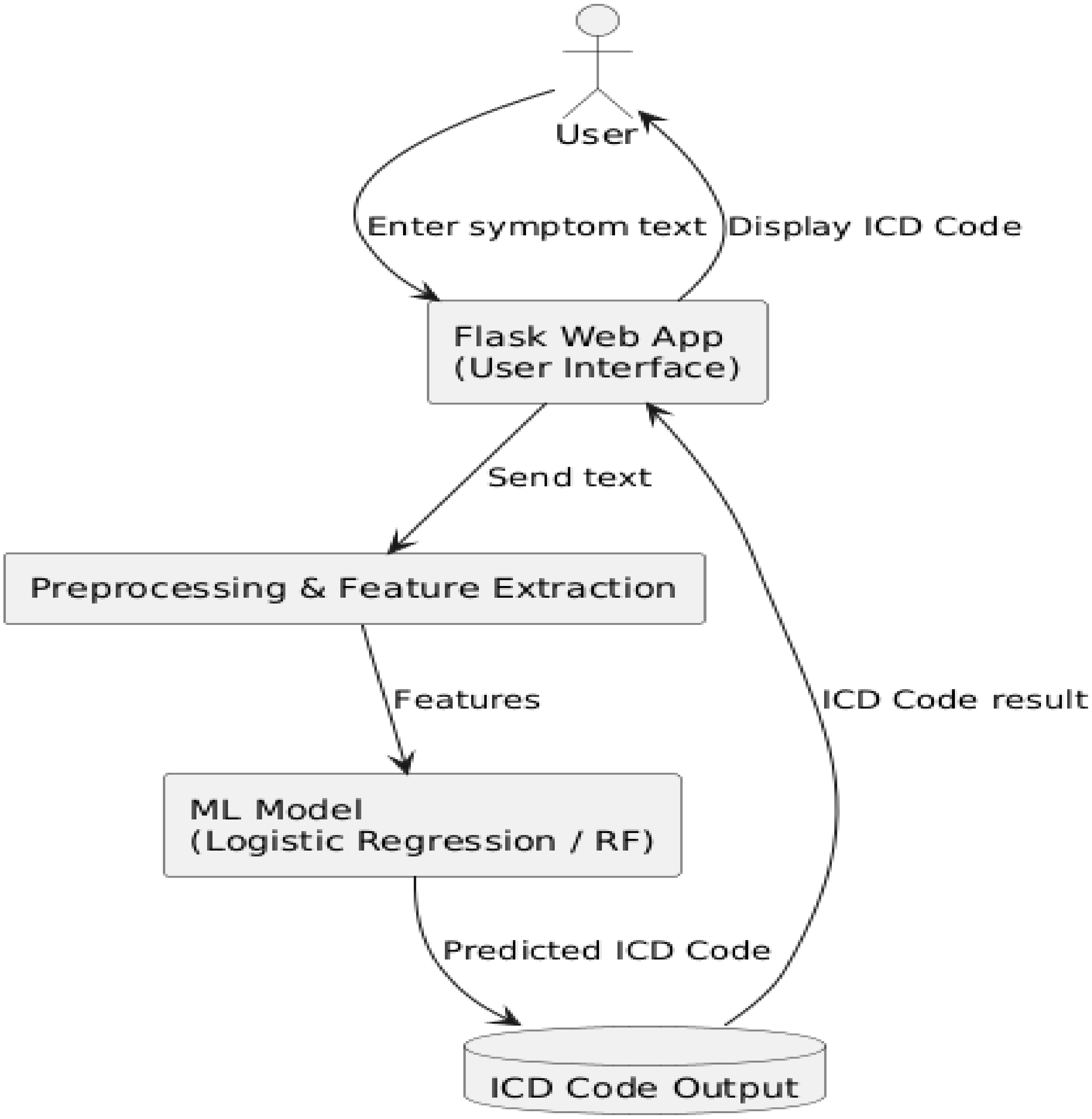


Fig 3.1: System Architecture

3.3 DEVELOPMENTAL ENVIRONMENT

3.3.1 HARDWARE REQUIREMENTS

The hardware specifications serve as a basis for the implementation and smooth functioning of the ICD Code Prediction System. A full description of hardware requirements is crucial for supporting both the model training phase and the web application deployment. These specifications guide the system design and ensure reliable performance throughout the project lifecycle.

Table 3.1 Hardware Requirements

COMPONENTS	SPECIFICATION
PROCESSOR	Intel Core i3
RAM	4 GB RAM
POWER SUPPLY	+5V power supply

3.3.2 SOFTWARE REQUIREMENTS

The software requirements define the essential components needed for developing, testing, and deploying the ICD Code Prediction System. These specifications guide the software engineers in planning, task management, cost estimation, and progress tracking throughout the development cycle.

Table 3.2 Software Requirements

COMPONENTS	SPECIFICATION
Operating System	Windows 7 or higher
Frontend	HTML ,CSS
Backend	Flask (Python)
Database	SQLite

3.4 DESIGN OF THE ENTIRE SYSTEM

3.4.1 ACTIVITY DIAGRAM

The activity diagram (Fig 3.2) represents the workflow for predicting ICD codes using a Flask-based machine learning system. The process begins with the user interacting via a web interface, where they provide a symptom description as input. The Flask framework serves as the backend, handling incoming user requests via a WSGI server. The user-submitted symptom text is then passed through the preprocessing module,where tasks such as text cleaning, lowercasing, stopword removal, tokenization, and lemmatization are performed to normalize the input. These preprocessed text features are converted into numerical vectors using TF-IDF vectorization.

The feature vectors are passed to the trained machine learning classification model, which processes the input using algorithms like Logistic Regression or Random Forest to predict the most appropriate ICD code. The model uses patterns learned from the labeled dataset to map the symptoms accurately to their corresponding ICD code.

Finally, the predicted ICD code is delivered back to the user through the web interface as output. This streamlined workflow ensures accurate, efficient, and user-friendly ICD code prediction, supporting healthcare professionals in clinical documentation and diagnosis coding.

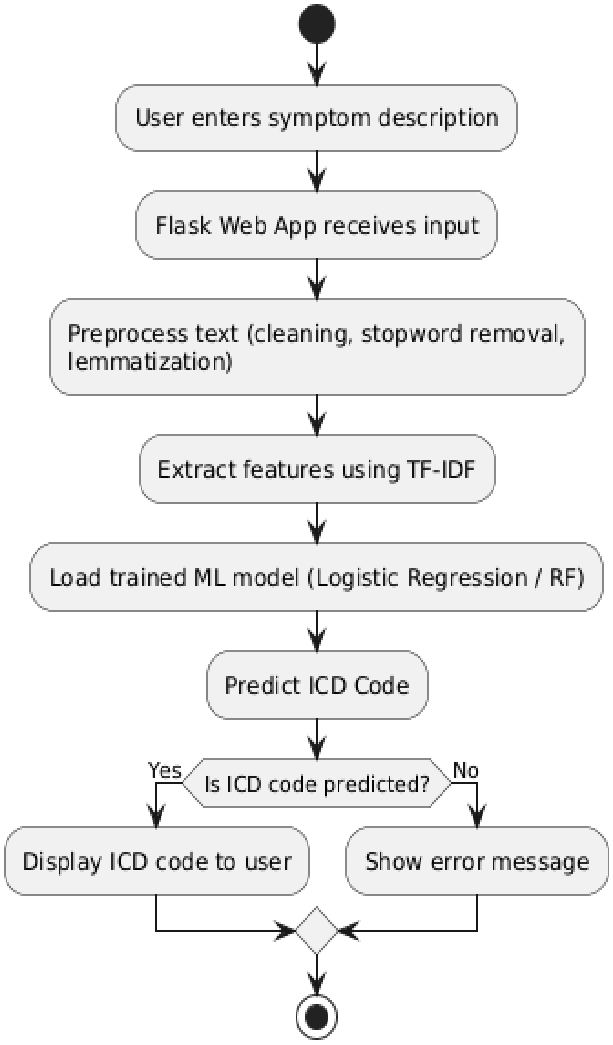

User enters symptom description
Flask Web App receives input
Preprocess text (cleaning, stopword removal, lemmatization)
Extract features using TF-IDF
Load trained ML model (Logistic Regression / RF)
Predict ICD Code
Yes
Is ICD code predicted?
No
Display ICD code to user
Show error message

Fig 3.2: Activity Diagram

3.4.2 DATA FLOW DIAGRAM

The data flow diagram **(Fig 3.3)** outlines the process of predicting ICD codes using a machine learning model deployed via a **Flask framework**. The system workflow begins with a **curated dataset**, containing symptom descriptions labeled with their corresponding ICD codes.

This raw dataset undergoes **preprocessing**, which includes text cleaning, lowercasing, stopword removal, tokenization, lemmatization, and TF-IDF vectorization to convert symptom descriptions into numerical feature vectors suitable for machine learning.

The preprocessed data is then **split** into **training data (80%)** and **testing data (20%)**.

The **training phase** involves fitting machine learning algorithms such as **Logistic Regression** and **Random Forest** on the training data to learn the patterns and relationships between symptoms and ICD codes.

Once trained, the model is **deployed** via the **Flask web framework**, providing a secure and scalable interface for real-time ICD code predictions. The **testing phase** evaluates the model's accuracy and robustness using performance metrics such as **accuracy**, **confusion matrix**, and **classification report**.

Finally, the system processes user-provided symptom descriptions via the web interface, and the trained model predicts and returns the most appropriate ICD code as output. This structured and efficient pipeline ensures accurate, reliable, and user-friendly ICD code prediction to support healthcare documentation and diagnosis coding.

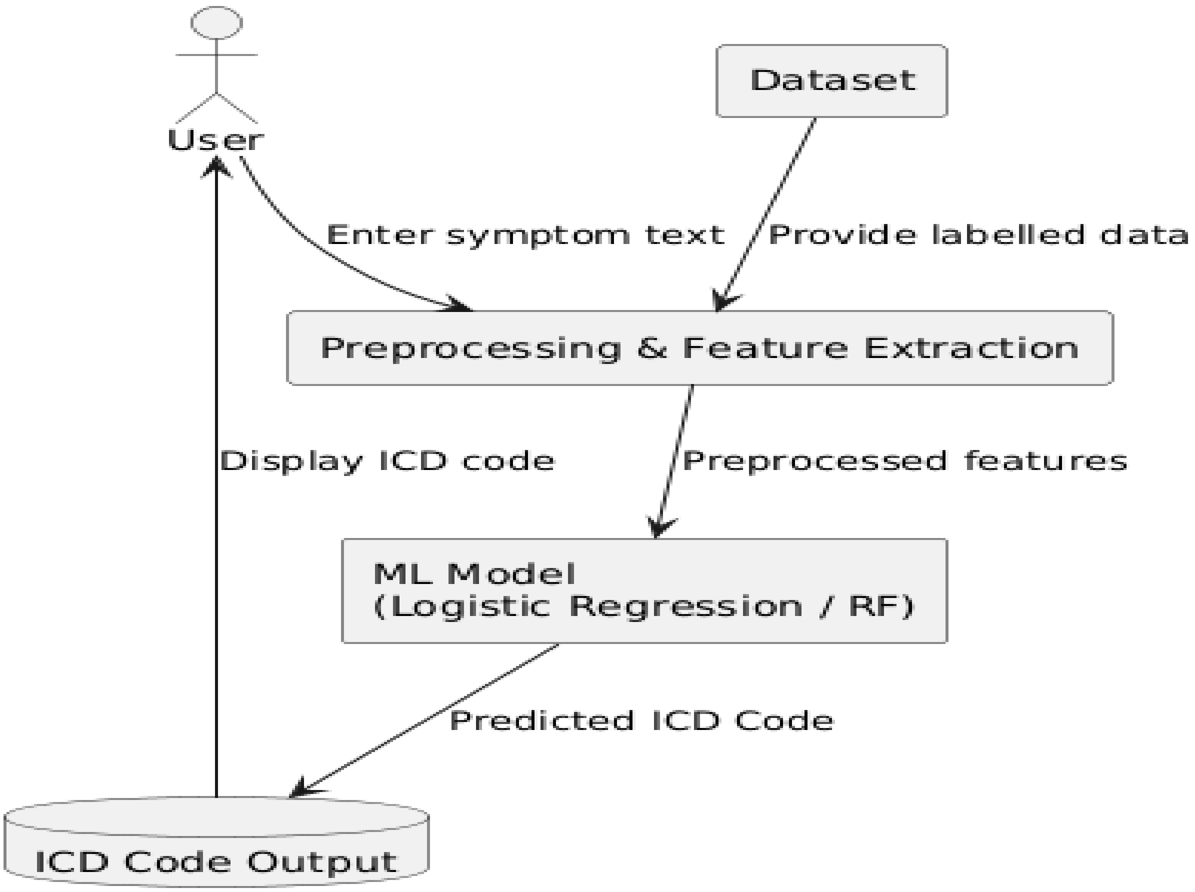


Fig 3.3:Data Flow Diagram

3.5 STATISTICAL ANALYSIS

The feature comparison table highlights the key differences between **traditional manual ICD coding systems** and the **proposed ICD Code Prediction System using Machine Learning**. The proposed system integrates advanced machine learning techniques, including automated feature extraction, optimized classification algorithms, and real-time prediction deployment, ensuring a more accurate, efficient, and scalable ICD coding process.While some features overlap with existing systems, the combination of **Natural Language Processing (NLP)** and **machine learning models** significantly enhances prediction accuracy, reduces manual errors, and streamlines the overall medical coding workflow.

Table 3.3 Comparison of features

Aspect	Existing System	Proposed System	Expected Outcomes
ICD Code Assignment	Manual coding by healthcare professionals	AI-powered prediction using Logistic Regression / Random Forest	Higher accuracy, reduced manual workload
Data Preprocessing	Minimal or no text preprocessing	Comprehensive text cleaning, stop word removal, lemmatization, and TF-IDF vectorization	Improved data quality for training and prediction
Feature Selection	Manual keyword identification	Automated feature extraction using TF-IDF	Optimized feature set for enhanced model performance
Performance Optimization	Limited manual validation	Iterative model tuning and evaluation using accuracy, confusion matrix, and classification report	Maximized prediction capabilities and system robustness

Deployment	Manual code search and entry	Flask-based automated ICD prediction system	Real-time, scalable ICD code predictions
Scalability	Limited to specific healthcare environments	Adaptable to diverse clinical datasets and settings	Enhanced flexibility and scalability in healthcare operations

The **Translating Medical Notes into ICD-10 Codes** stands out through its innovative features, distinguishing it from traditional manual ICD coding methods. Notably, it integrates advanced machine learning techniques such as **Logistic Regression**, **Random Forest**, and **TF-IDF vectorization** to enhance prediction accuracy and reliability.Additionally, the system ensures efficient data preprocessing, enabling high-quality feature extraction and optimized model performance. Improved scalability and efficiency are key advantages, allowing the system to process large volumes of symptom descriptions in real time with minimal manual intervention.The **user-friendly web interface** developed using Flask ensures intuitive interaction for users, making ICD code prediction seamless and accessible. By significantly reducing manual errors, enhancing trust and accuracy, and providing real-time prediction results, the system effectively supports healthcare professionals in clinical documentation and diagnosis coding.While traditional systems may offer some of these features individually, the **holistic approach** of the ICD Code Prediction System ensures a comprehensive and robust solution for automating ICD code assignment.

CHAPTER 4

MODULE DESCRIPTION

The workflow for the proposed system is designed to ensure a structured and efficient process for predicting ICD codes from clinical symptom descriptions. It consists of the following sequential steps:

4.1 SYSTEM ARCHITECTURE

4.1.1 USER INTERFACE DESIGN

The sequence diagram (Fig 4.1) depicts the process of predicting ICD codes. It starts with the user providing the input — a free-text description of clinical symptoms — through a web-based form.

The input text is passed to the Flask backend server, where it undergoes text preprocessing operations such as cleaning, stopword removal, lemmatization, and TF-IDF feature extraction.

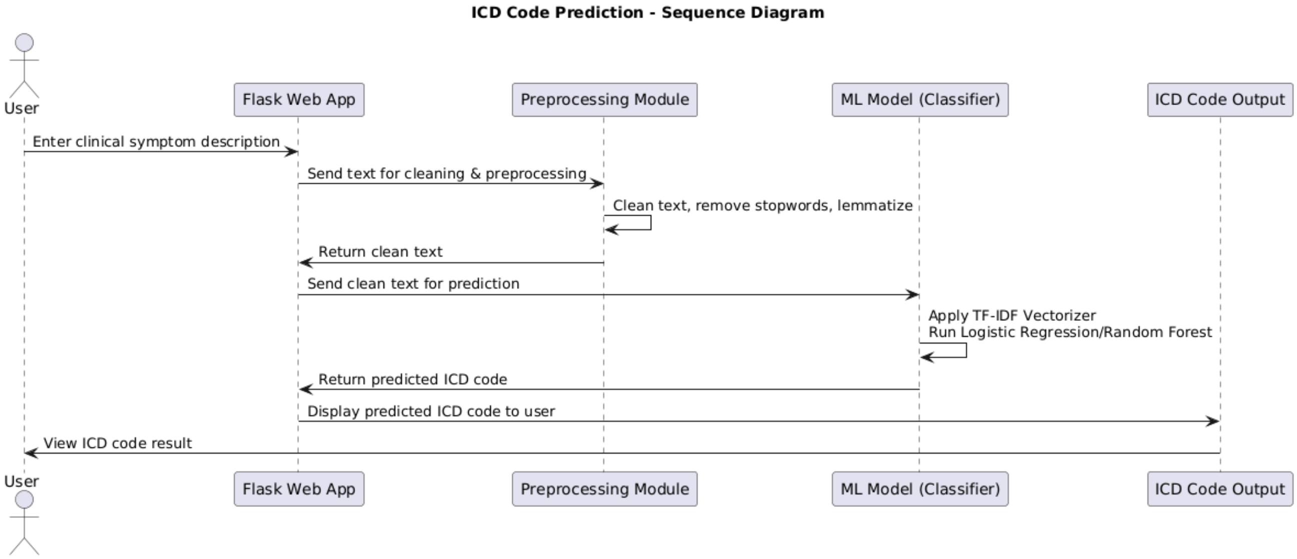

ICD Code Prediction - Sequence Diagram
User
Flask Web App
Preprocessing Module
ML Model (Classifier)
ICD Code Output
Enter clinical symptom description
Send text for cleaning & preprocessing
Clean text, remove stopwords, lemmatize
Return clean text
Send clean text for prediction
Apply TF-IDF Vectorizer
Run Logistic Regression/Random Forest
Return predicted ICD code
Display predicted ICD code to user
View ICD code result
User
Flask Web App
Preprocessing Module
ML Model (Classifier)
ICD Code Output

Fig 4.1: SEQUENCE DIAGRAM

4.1.2 BACK END INFRASTRUCTURE

The backend infrastructure for the sequence diagram comprises a centralized database for managing datasets, storing raw and labeled clinical data for preprocessing, training, and testing. A machine learning framework such as Scikit-learn is used to implement and train models like Logistic Regression and Random Forest.

The Flask framework with a WSGI server handles API requests and application logic, enabling seamless interaction between users and the backend for predictions and result delivery. The centralized server manages communication between the web interface and the prediction engine, ensuring secure and efficient processing.

4.2 DATA COLLECTION AND PREPROCESSING

4.2.1 Dataset and Data Labelling

Labeled datasets are collected, including symptom descriptions paired with corresponding ICD codes. Accurate labeling differentiates between various clinical conditions and ensures effective model training.

4.2.2. Data Preprocessing

The raw dataset undergoes extensive preprocessing, which includes:

- Data Cleaning: Removal of inconsistent, redundant, or irrelevant data
- Missing Value Replacement: Imputation techniques to handle incomplete entries
- Text Normalization: Lowercasing, stopword removal, tokenization, and lemmatization to standardize text data.

4.2.3 Feature Selection

Advanced techniques are used to ensure relevant and optimized feature sets:

- **TF-IDF Vectorization**: Converts text into numerical feature vectors based on term frequency and inverse document frequency
- **Dimensionality Reduction**: Reducing data complexity while retaining critical features for accurate classification

4.2.4 Classification and Model Selection

Multiple models are evaluated for classification, such as:

- Logistic Regression: For multi-class text classification tasks
- Random Forest: For general-purpose and robust classification
- Model Selection: The best-performing model (Logistic Regression or Random Forest) is selected based on accuracy and adaptability in ICD code prediction

4.2.5 Performance Evaluation and Optimization

Model performance is assessed using metrics such as accuracy, confusion matrix, and classificationreport.The chosen model undergoes iterative tuning and optimization to maximize prediction accuracy and reduce misclassifications.

4.2.6 Model Deployment

The optimized model is deployed via a Flask-based system, enabling real-time ICD code predictions by processing live user inputs through the web application interface.

4.2.7 Centralized Server and Database

All data, including training results, predictions, and evaluations, is securely stored in acentralizeddatabase.The server handles communication between the web interface and the machine learning model, ensuring secure data processing.

4.3 SYSTEM WORK FLOW

4.3.1 User Interaction:

Users initiate the prediction process by submitting symptom descriptions through the webapplicationinterface.The system processes these inputs and prepares them for classification.

4.3.2 ICD Code Prediction:

Advanced machine learning techniques (Logistic Regression and Random Forest) are applied to identify patterns associated with various clinical conditions. The system analyzes the symptom description and predicts the most relevant ICD code.

4.3.3 Result Display and Reporting:

Once the ICD code is predicted, it is displayed to the user in real time via the web interface.The result provides healthcare professionals with accurate and standardized diagnostic codes for documentation purposes.

4.3.4 Continuous Learning and Improvement:

The system can continuously update and retrain its machine learning models based on newsymptompatternsanduserFeedback. This ensures the model remains accurate and adaptable to evolving clinical terminologies and emerging healthcare needs.

CHAPTER 5

IMPLEMENTATION AND RESULTS

5.1 IMPLEMENTATION

The project is developed and deployed using a robust technology stack, consisting of Python for backend processing, Flask as the web framework, and SQLite for simple and effective database management. The frontend is designed using HTML and CSS, ensuring a responsive and user-friendly interface.

For ICD code prediction, the system leverages machine learning algorithms, including Logistic Regression and Random Forest, to analyze clinical symptom descriptions and predict the corresponding ICD codes accurately. The implementation involves setting up an intuitive web interface, allowing healthcare professionals to submit symptom descriptions for ICD code assignment.

The backend server efficiently processes user inputs, performs text preprocessing (including cleaning, lemmatization, and TF-IDF vectorization), and passes the processed data to the trained machine learning model for prediction. The predicted ICD code is displayed to the user in real-time through the web interface.

Additionally, a data management system is implemented, enabling storage and retrieval of symptom–ICD code pairs, model outputs, and evaluation metrics. The system is designed to be adaptable and scalable, allowing future updates with new clinical terms and improved prediction models. Continuous learning and model retraining ensure improved accuracy and reliability over time.

5.2 OUTPUT SCREENSHOTS

The project implementation is structured into modules and is demonstrated through a series of outputs:

- Fig 5.1 depicts the project’s overall system workflow, highlighting the seamless integration of machine learning for predictive analysis. It demonstrates the clear flow from data collection to real-time ICD code prediction through the web interface, ensuring usability and accessibility.

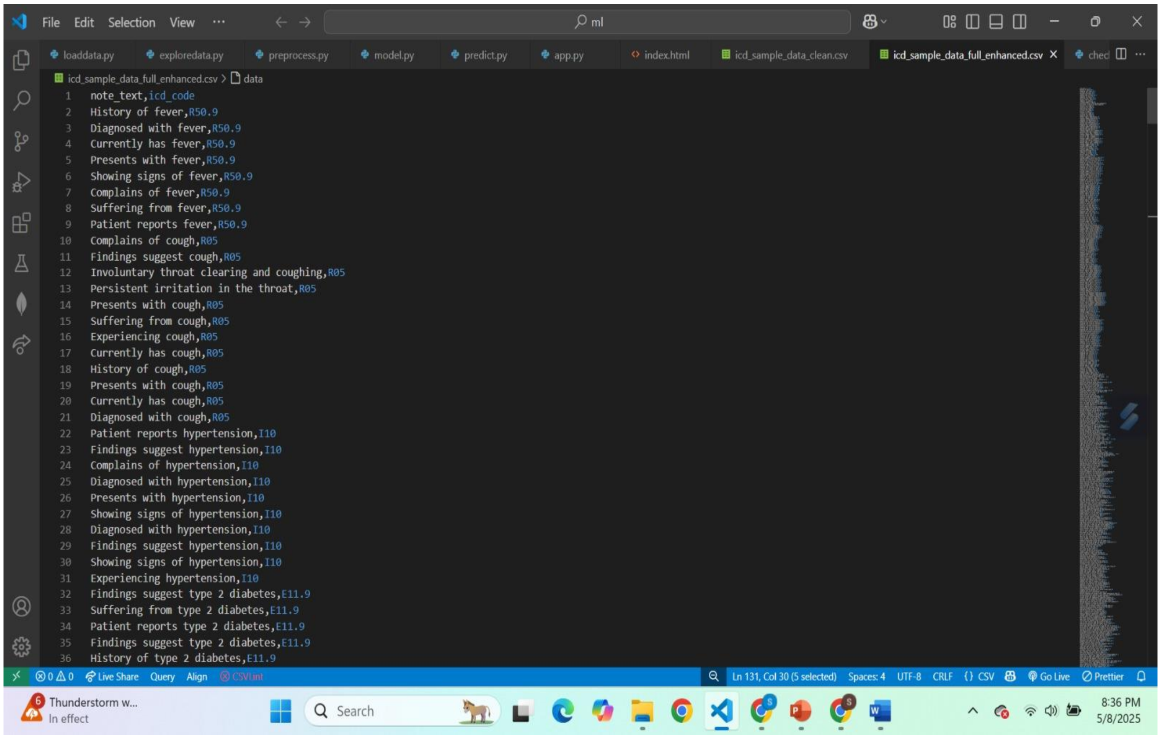
- Fig 5.2 showcases the machine learning model development and training process. It illustrates how clinical symptom descriptions are preprocessed, feature-extracted using TF-IDF, and fed into classifiers like Logistic Regression and Random Forest for model training and evaluation.

- Fig 5.3 compares the confusion matrices of multiple classifiers. It highlights the models' performance in predicting ICD codes, showing trends in accuracy and misclassifications. This visualization aids in selecting the best-performing algorithm for deployment.

- Fig 5.4 demonstrates the integration of the trained machine learning model within a Flask web application. The interface accepts user input of symptom descriptions and returns the predicted ICD code in real-time, ensuring effective deployment for practical healthcare use cases.

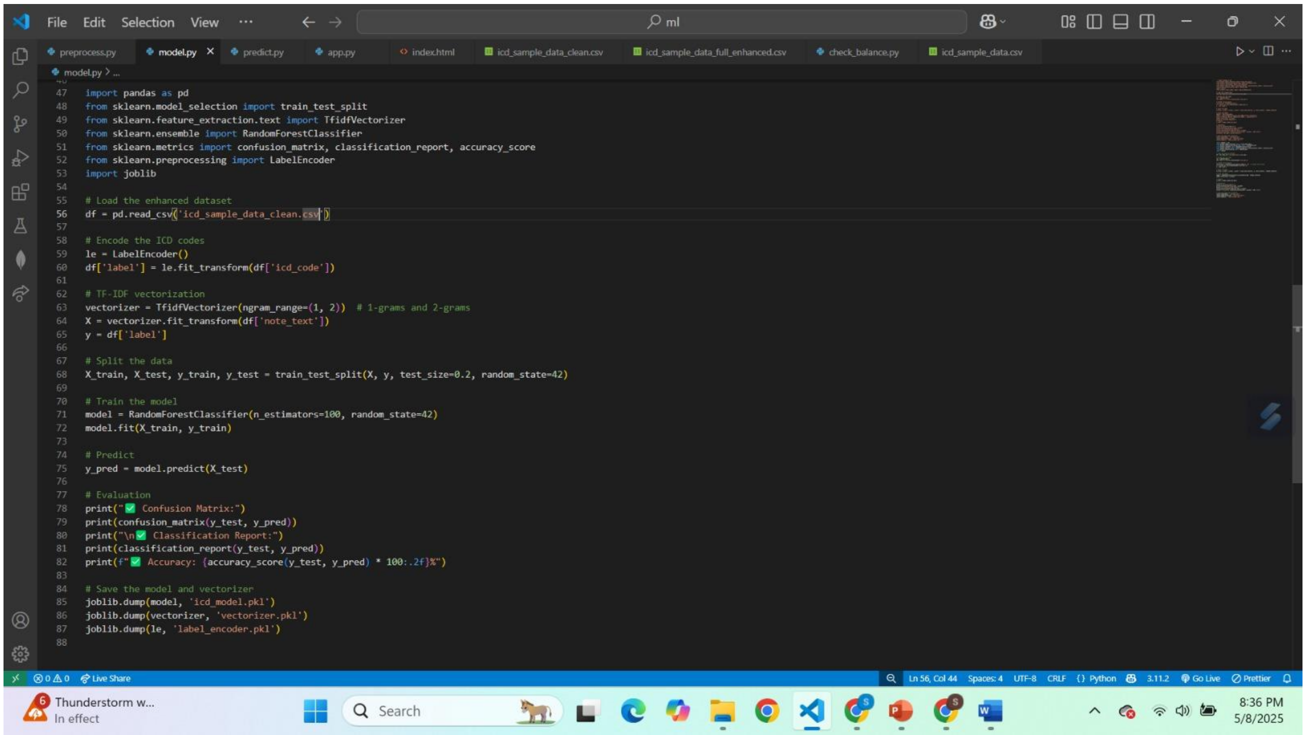
- Fig 5.5 illustrates the Flask web application’s user interface designed for ICD code prediction. The interface is simple and user-friendly, allowing healthcare professionals to input symptoms and receive ICD predictions quickly and efficiently.

- Fig 5.6 presents the prediction result page of the Flask web application. It displays the predicted ICD code along with a confidence score or classification outcome. The page includes a "Go Back" button for easy navigation, offering a seamless and smooth user experience**.**


File Edit Selection View
loaddata.py
exploredata.py
preprocess.py
model.py
predict.py
app.py
index.html
icd_sample_data_clean.csv
icd_sample_data_full_enhanced.csv
icd_sample_data_full_enhanced.csv > data
Ln 131, Col 30 (5 selected) Spaces: 4 UTF-8 CRLF CSV
Thunderstorm w... In effect
8:36 PM 5/8/2025

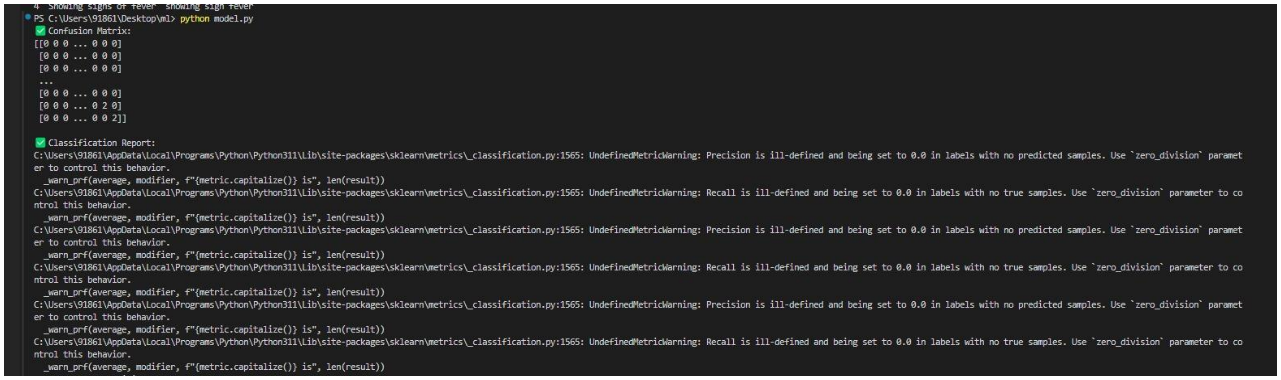
```
note_text,icd_code
History of fever,R50.9
Diagnosed with fever,R50.9
Currently has fever,R50.9
Presents with fever,R50.9
Showing signs of fever,R50.9
Complains of fever,R50.9
Suffering from fever,R50.9
Patient reports fever,R50.9
Complains of cough,R05
Findings suggest cough,R05
Involuntary throat clearing and coughing,R05
Persistent irritation in the throat,R05
Presents with cough,R05
Suffering from cough,R05
Experiencing cough,R05
Currently has cough,R05
History of cough,R05
Presents with cough,R05
Currently has cough,R05
Diagnosed with cough,R05
Patient reports hypertension,I10
Findings suggest hypertension,I10
Complains of hypertension,I10
Diagnosed with hypertension,I10
Presents with hypertension,I10
Showing signs of hypertension,I10
Diagnosed with hypertension,I10
Findings suggest hypertension,I10
Showing signs of hypertension,I10
Experiencing hypertension,I10
Findings suggest type 2 diabetes,E11.9
Suffering from type 2 diabetes,E11.9
Patient reports type 2 diabetes,E11.9
Findings suggest type 2 diabetes,E11.9
History of type 2 diabetes,E11.9
```

Fig 5.1 Dataset for Training



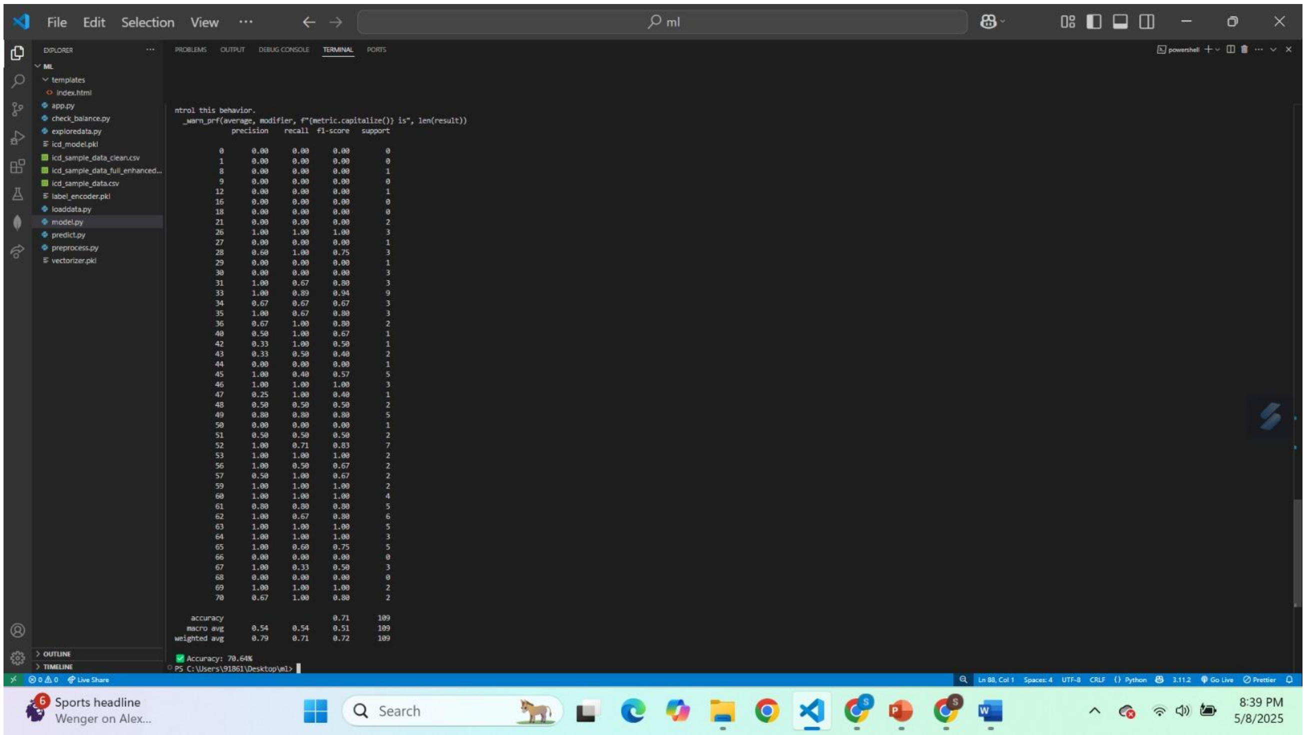
```python
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import confusion_matrix, classification_report, accuracy_score
from sklearn.preprocessing import LabelEncoder
import joblib

# Load the enhanced dataset
df = pd.read_csv('icd_sample_data_clean.csv')

# Encode the ICD codes
le = LabelEncoder()
df['label'] = le.fit_transform(df['icd_code'])

# TF-IDF vectorization
vectorizer = TfidfVectorizer(ngram_range=(1, 2))  # 1-grams and 2-grams
X = vectorizer.fit_transform(df['note_text'])
y = df['label']

# Split the data
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# Train the model
model = RandomForestClassifier(n_estimators=100, random_state=42)
model.fit(X_train, y_train)

# Predict
y_pred = model.predict(X_test)

# Evaluation
print("✅ Confusion Matrix:")
print(confusion_matrix(y_test, y_pred))
print("\n✅ Classification Report:")
print(classification_report(y_test, y_pred))
print(f"✅ Accuracy: {accuracy_score(y_test, y_pred) * 100:.2f}%")

# Save the model and vectorizer
joblib.dump(model, 'icd_model.pkl')
joblib.dump(vectorizer, 'vectorizer.pkl')
joblib.dump(le, 'label_encoder.pkl')
```

Fig 5.2 Performance Evaluation & Optimization



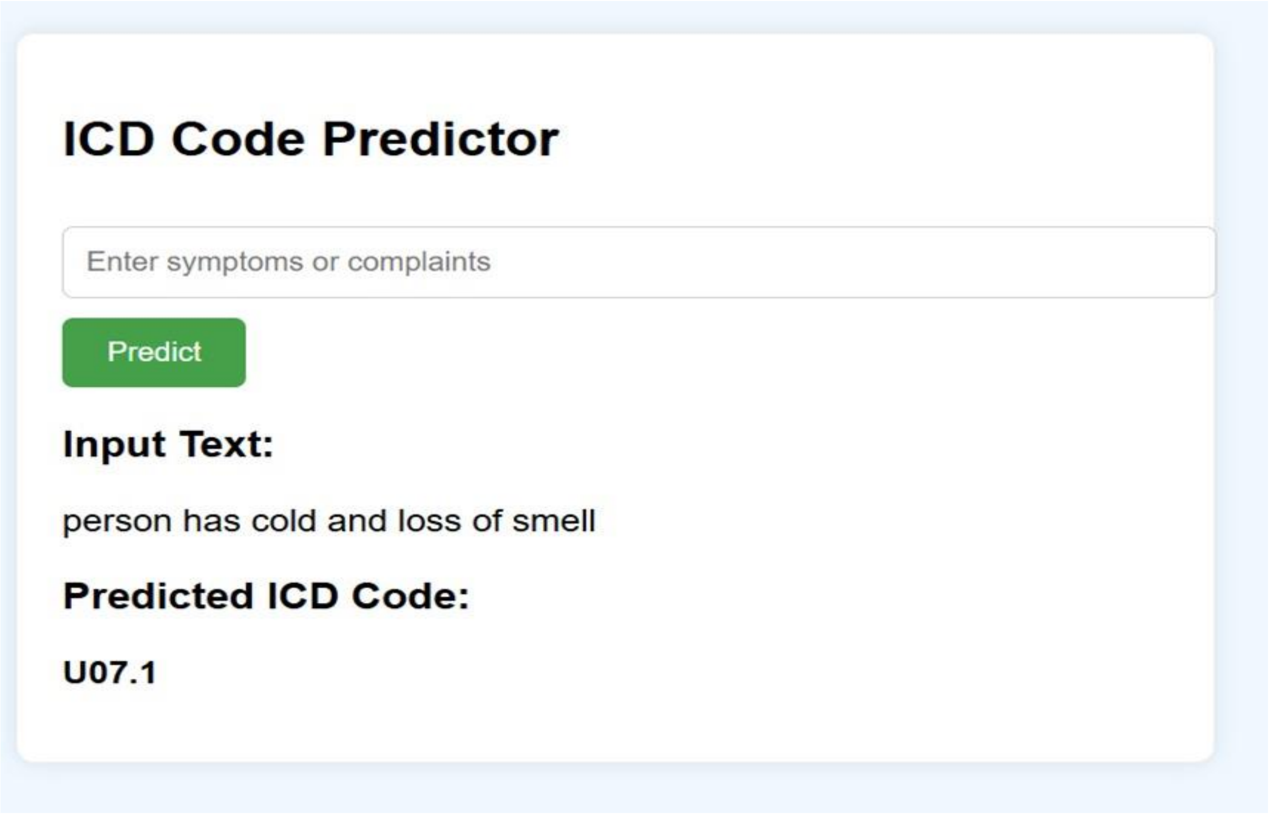
```
4  Showing signs of fever  showing sign fever
PS C:\Users\91861\Desktop\ml> python model.py
✅ Confusion Matrix:
[[0 0 0 ... 0 0 0]
 [0 0 0 ... 0 0 0]
 [0 0 0 ... 0 0 0]
 ...
 [0 0 0 ... 0 0 0]
 [0 0 0 ... 0 2 0]
 [0 0 0 ... 0 0 2]]

✅ Classification Report:
C:\Users\91861\AppData\Local\Programs\Python\Python311\Lib\site-packages\sklearn\metrics\_classification.py:1565: UndefinedMetricWarning: Precision is ill-defined and being set to 0.0 in labels with no predicted samples. Use `zero_division` paramet
er to control this behavior.
  _warn_prf(average, modifier, f"{metric.capitalize()} is", len(result))
C:\Users\91861\AppData\Local\Programs\Python\Python311\Lib\site-packages\sklearn\metrics\_classification.py:1565: UndefinedMetricWarning: Recall is ill-defined and being set to 0.0 in labels with no true samples. Use `zero_division` parameter to co
ntrol this behavior.
  _warn_prf(average, modifier, f"{metric.capitalize()} is", len(result))
C:\Users\91861\AppData\Local\Programs\Python\Python311\Lib\site-packages\sklearn\metrics\_classification.py:1565: UndefinedMetricWarning: Precision is ill-defined and being set to 0.0 in labels with no predicted samples. Use `zero_division` paramet
er to control this behavior.
  _warn_prf(average, modifier, f"{metric.capitalize()} is", len(result))
C:\Users\91861\AppData\Local\Programs\Python\Python311\Lib\site-packages\sklearn\metrics\_classification.py:1565: UndefinedMetricWarning: Recall is ill-defined and being set to 0.0 in labels with no true samples. Use `zero_division` parameter to co
ntrol this behavior.
  _warn_prf(average, modifier, f"{metric.capitalize()} is", len(result))
C:\Users\91861\AppData\Local\Programs\Python\Python311\Lib\site-packages\sklearn\metrics\_classification.py:1565: UndefinedMetricWarning: Precision is ill-defined and being set to 0.0 in labels with no predicted samples. Use `zero_division` paramet
er to control this behavior.
  _warn_prf(average, modifier, f"{metric.capitalize()} is", len(result))
C:\Users\91861\AppData\Local\Programs\Python\Python311\Lib\site-packages\sklearn\metrics\_classification.py:1565: UndefinedMetricWarning: Recall is ill-defined and being set to 0.0 in labels with no true samples. Use `zero_division` parameter to co
ntrol this behavior.
  _warn_prf(average, modifier, f"{metric.capitalize()} is", len(result))
```

Fig 5.3 Confusion Matrix



```
ntrol this behavior.
  _warn_prf(average, modifier, f"{metric.capitalize()} is", len(result))
              precision    recall  f1-score   support

           0       0.00      0.00      0.00         0
           1       0.00      0.00      0.00         0
           8       0.00      0.00      0.00         1
           9       0.00      0.00      0.00         0
          12       0.00      0.00      0.00         1
          16       0.00      0.00      0.00         0
          18       0.00      0.00      0.00         0
          21       0.00      0.00      0.00         2
          26       1.00      1.00      1.00         3
          27       0.00      0.00      0.00         1
          28       0.60      1.00      0.75         3
          29       0.00      0.00      0.00         1
          30       0.00      0.00      0.00         3
          31       1.00      0.67      0.80         3
          33       1.00      0.89      0.94         9
          34       0.67      0.67      0.67         3
          35       1.00      0.67      0.80         3
          36       0.67      1.00      0.80         2
          40       0.50      1.00      0.67         1
          42       0.33      1.00      0.50         1
          43       0.33      0.50      0.40         2
          44       0.00      0.00      0.00         1
          45       1.00      0.40      0.57         5
          46       1.00      1.00      1.00         3
          47       0.25      1.00      0.40         1
          48       0.50      0.50      0.50         2
          49       0.80      0.80      0.80         5
          50       0.00      0.00      0.00         1
          51       0.50      0.50      0.50         2
          52       1.00      0.71      0.83         7
          53       1.00      1.00      1.00         2
          56       1.00      0.50      0.67         2
          57       0.50      1.00      0.67         2
          59       1.00      1.00      1.00         2
          60       1.00      1.00      1.00         4
          61       0.80      0.80      0.80         5
          62       1.00      0.67      0.80         6
          63       1.00      1.00      1.00         5
          64       1.00      1.00      1.00         3
          65       1.00      0.60      0.75         5
          66       0.00      0.00      0.00         0
          67       1.00      0.33      0.50         3
          68       0.00      0.00      0.00         0
          69       1.00      1.00      1.00         2
          70       0.67      1.00      0.80         2

    accuracy                           0.71       109
   macro avg       0.54      0.54      0.51       109
weighted avg       0.79      0.71      0.72       109

✅ Accuracy: 70.64%
PS C:\Users\91861\Desktop\ml>
```

Fig 5.4 ICD Code Prediction



ICD Code Predictor

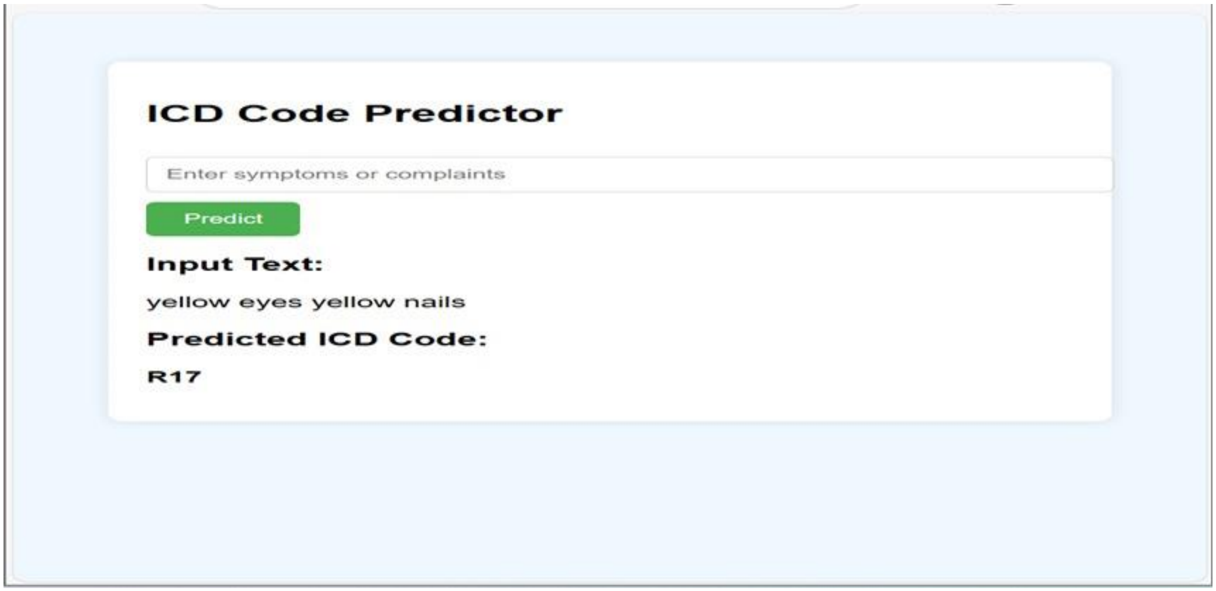
Input Text:

person has cold and loss of smell

Predicted ICD Code:

U07.1

Fig 5.5 ICD Prediction from medical notes



ICD Code Predictor

Input Text:

yellow eyes yellow nails

Predicted ICD Code:

R17

Fig 5.6 Prediction result

CHAPTER 6

CONCLUSION AND FUTURE ENHANCEMENT

6.1 CONCLUSION

The proposed system leverages **machine learning** and **natural language processing (NLP)** technologies to create an innovative solution for automating **ICD code prediction** from clinical symptom descriptions, effectively addressing challenges in manual medical coding with enhanced accuracy, efficiency, and reliability.By analyzing a broad set of clinical text features, machine learning enables precise prediction of appropriate ICD codes, adapting to varying patterns and terminology in clinical narratives. The use of techniques such as **TF-IDF vectorization**, **Logistic Regression**, and **Random Forest** ensures robust feature extraction and accurate classification, supporting healthcare professionals in streamlining the coding process.The system's user-friendly, **Flask-based web application** allows seamless deployment, providing healthcare providers with a real-time tool to assign ICD codes efficiently. By automating the coding process, the platform reduces manual workload, minimizes errors, and enhances the quality and consistency of healthcare documentation.

6.2 FUTURE ENHANCEMENT

Future enhancements for this project could include integrating **deep learning models** such as **Convolutional Neural Networks (CNNs)** and **transformers like BERT** for more advanced and context-aware text analysis of complex clinical symptom descriptions. The use of **pre-trained biomedical language models** can further improve the system's ability to understand domain-specific medical terminologies, thereby enhancing prediction accuracy.Incorporating **transfer learning** techniques can enable the system to adapt quickly to new clinical datasets and evolving medical terminologies without extensive retraining.

REFERENCES

[1] Agarwal, S., and Yu, H. "Automatically classifying sentences in full-text biomedical articles into introduction, methods, results and discussion." *Bioinformatics* 25, no. 23 (2009): 3174-3180.

[2] Koopman, B., Karimi, S., Nguyen, A., McGuire, R., Muscatello, D., Kemp, M., and Truran, D. "Automatic classification of diseases from free-text death certificates for real-time surveillance." *BMC Medical Informatics and Decision Making* 15, no. 1 (2015): 53.

[3] Perotte, A., Pivovarov, R., Natarajan, K., Weiskopf, N., Wood, F., and Hripcsak, G. "Diagnosis code assignment: models and evaluation metrics." *Journal of the American Medical Informatics Association* 21, no. 2 (2014): 231-237.

[4] Mullenbach, J., Wiegreffe, S., Duke, J., Sun, J., and Eisenstein, J. "Explainable prediction of medical codes from clinical text." In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics*, pp. 1101-1111. 2018.

[5] Shi, X., Zhang, J., Sun, J., and Zhang, Q. "Towards automated ICD coding using deep learning." In *Proceedings of the 2017 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, pp. 994–997. IEEE, 2017.

[6] Xu, Y., Zhang, Y., Wang, J., Tang, J., and Zhang, J. "Multi-label classification of medical texts using deep learning." *Journal of Biomedical Informatics* 93 (2019): 103141.

[7] Liu, Z., Chen, C., Wang, Y., and Liu, W. "Automatic ICD coding using deep learning: a systematic review." *Computer Methods and Programs in Biomedicine* 212 (2022): 106473.

[8] Pivovarov, R., Perotte, A., Grave, E., Angiolillo, J., and Hripcsak, G. "Learning probabilistic phenotypes from heterogeneous EHR data." *Journal of Biomedical Informatics* 58 (2015): 156-165.

[9] Kavuluru, R., Rios, A., and Lu, Y. "An empirical evaluation of supervised learning

approaches in assigning diagnosis codes to electronic medical records." *Artificial Intelligence in Medicine* 78 (2017): 1–8.

[10] Wang, Y., Wang, L., Rastegar-Mojarad, M., Moon, S., Shen, F., Afzal, N., Liu, S., and Liu, H. "Clinical information extraction applications: a literature review." *Journal of Biomedical Informatics* 77 (2018): 34–49.