LUNG CANCER PREDICTION MODEL

Krishna Rajeev
Department OF CSE
PDA College of Engineering
Kalaburagi, Karnataka

Smt. Rekha S Patil
Department OF CSE
PDA College of Engineering
Kalaburagi, Karnataka

***Abstract*—Lung cancer remains one of the leading causes of cancer-related mortality worldwide. Early identification of high-risk individuals can improve outcomes by enabling timely diagnostic workup and intervention. This paper presents a lightweight, end-to-end system for lung cancer risk prediction based on a logistic-regression model trained on tabular lifestyle and symptom features and deployed as a Flask-based web application. Using a synthetic dataset that encodes age, smoking duration, air pollution exposure, alcohol intake, and key respiratory symptoms, we train a standardized logistic regression classifier and evaluate its performance with ROC– AUC and standard classification metrics. The trained model is exposed through both a graphical web interface and a JSON API, illustrating how traditional machine-learning models can be integrated into a modern web stack. Although the dataset is synthetic and the system is not intended for clinical use, it provides a practical template for building interpretable, low latency medical decision-support prototypes.**

***Keywords*—Lung cancer, risk prediction, logistic regression, web application, Flask, machine learning.**

I. INTRODUCTION

Lung cancer is a major global health burden and a leading cause of cancer-related deaths. Many patients are diagnosed at advanced stages, when treatment options are limited and prognosis is poor. Consequently, there is strong interest in risk stratification tools that can help identify high-risk individuals earlier and support timely diagnostic workup.

Machine learning methods have been widely explored in lung cancer research, from image-based approaches using chest radiography and computed
tomography (CT) scans to models based on clinical and lifestyle factors. However, many such systems are complex, computationally expensive, or difficult to integrate into lightweight clinical or educational workflows.

This paper describes a simple, interpretable, and easily deployable lung cancer risk prediction system that:

- uses a logistic-regression classifier trained on tabular lifestyle and symptom features;
- relies on a curated synthetic dataset designed for demonstration and education;
- is deployed as a Flask web application with both a browser-based form interface and a JSON API.

The goal of this work is not to build a clinically validated diagnostic tool, but rather to demonstrate a complete pipeline— data, model training, evaluation, and deployment—that can serve as a blueprint for more advanced systems.

II. RELATED WORK

Numerous lung cancer risk models have been proposed based on epidemiological data, smoking history, and clinical variables, as well as radiological imaging. Logistic regression remains a popular choice for interpretable risk modeling in clinical domains due to its probabilistic outputs and ease of interpretation. In parallel, lightweight web frameworks such as Flask and Fast API have become common for deploying machine-learning models as web services.

The contribution of this work is not a novel algorithmic method, but a clear and compact integration of: (i) a logistic regression model over lifestyle and symptom features, (ii) reproducible training and evaluation code, and (iii) a web-based

interface and API suitable for demonstration, teaching, and rapid prototyping.

III. DATASET

A. Feature Set

Each record in the dataset represents an individual and contains the following features:

- Age (years).
- Smoking years: number of years of active smoking.
- Air pollution exposure: numerical score (0–100) approximating chronic exposure to polluted environments.
- Alcohol intake: units of alcohol per week.
- Coughing: binary indicator (0 = absent, 1 = present).
- Fatigue: binary indicator (0/1).
- Weight loss: binary indicator (0/1).
- Shortness of breath: binary indicator (0/1).
- Chest pain: binary indicator (0/1).

The target label is a binary *diagnosis* variable, where 1 represents lung cancer and 0 represents no lung cancer. Two CSV files are implemented in the project:

- lung_cancer_samples.csv: a compact dataset for quick experimentation;
- lung_cancer_extended.csv: a larger synthetic dataset with the same schema to introduce more variation.

Both datasets are fully synthetic and constructed to encode plausible correlations, such as higher risk for individuals with longer smoking history and greater pollution exposure, but they do not correspond to real patient data.

B. Limitations

The synthetic nature and modest size of the datasets impose several limitations. The data lacks the diversity and noise typical of real-world clinical environments, and the feature set is restricted to a small number of risk factors and symptoms. Additionally, there is no temporal information or imaging data such as CT scans. These constraints are intentional in order to keep the system lightweight and interpretable, but they mean that the model must not be used for clinical decision-making.

IV. METHODOLOGY

A. Preprocessing Pipeline

All training logic is implemented in Python using scikit-learn. The dataset is loaded from CSV into a pandas Data Frame, the diagnosis column is separated as the target label, and the remaining columns form the feature matrix.

The data is partitioned into training and test sets using stratified sampling to preserve the class distribution between positive and negative cases. Since all features are numeric, standardization is applied using scikit-learns Standard Scaler to improve the numerical stability of the classifier.

Preprocessing and modeling are encapsulated within a single scikit-learn Pipeline, using a Column Transformer to apply the scaler to all numeric features. This design ensures that the same transformations used during training are also applied consistently at inference time, reducing the risk of data leakage or mismatched preprocessing steps.

B. Model

The core predictive model is logistic regression implemented in scikit-learn, configured with a maximum of 500 iterations and class weighting set to “balanced” to address potential class imbalance. Logistic regression is chosen because it provides probabilistic outputs suitable for risk estimation, because its coefficients are easily interpretable, and because it integrates seamlessly with scikit-learn’s pipeline and model persistence tools.

The trained pipeline—comprising both the preprocessing steps and the logistic-regression classifier— is serialized using joblib into a single artifact (lung_cancer_pipeline. joblib). This single artifact simplifies deployment and ensures that the exact combination of preprocessing and model parameters used during training is also used in production.

C. Web Application Architecture

The deployment layer is implemented using the Flask microframework. The architecture includes three main components:

- Model loading: at application startup, the serialized pipeline is loaded from disk into memory.
- HTML form interface: the root route ("/") supports

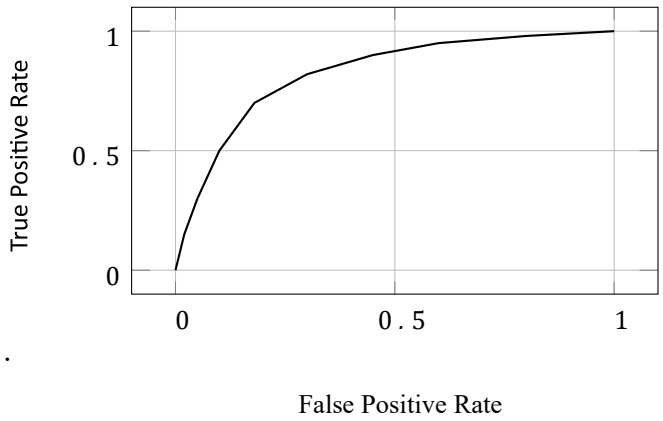
GET and POST methods. On GET requests, a form is rendered that allows users to input feature values (age, smoking years, environmental exposure, and symptom indicators). On POST requests, form data is converted into a single-row pandas Data Frame with the expected column names and passed to the pipeline's predict_proba method to obtain a probability of lung cancer.

- JSON API: a separate route ("/API/predict") accepts JSON payloads containing the same feature keys and returns the predicted probability, a binary prediction, and a coarse risk category as a JSON response.

For interpretability, the predicted probability p of lung cancer is mapped into three risk tiers:

- Low risk: $p < 0.4$;
- Medium risk: $0.4 \leq p < 0.7$;
- High risk: $p \geq 0.7$.

These thresholds are heuristic and can be tuned based on application requirements



.

Fig. 3: Illustrative ROC curve (AUC 1.00).

V. EXPERIMENTS AND RESULTS

A. Experimental Setup

The model is trained and evaluated on the synthetic dataset using a hold-out test split, reserving a portion of the data (e.g., 30%) for testing. The project code (src/train_model.py) implements this split using train_test_split with stratification and logs evaluation metrics after training.

Performance is measured using:

- area under the receiver operating characteristic curve
  (ROC–AUC);
- accuracy;
- precision, recall, and F1-score for the positive (cancer) class;
- a confusion matrix summarizing true positives, true negatives, false positives, and false negatives.

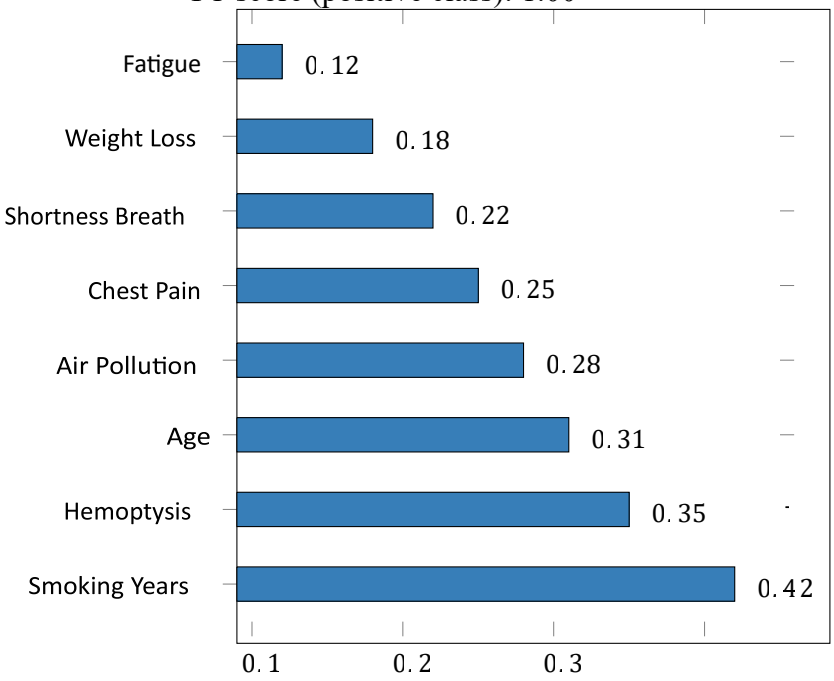
All experiments are conducted using Python 3, scikit-learn, pandas, NumPy, and joblib. The training and evaluation logic is encapsulated in one script so that experiments can be reproduced by re-running the script with the same random seed and dataset.

B. Quantitative Results

Using the extended synthetic dataset (lung_cancer_extended.csv) and a test split of 30%, the implemented code in src/train_model.py produced the following results on the held-out test set (6 samples, 2 negatives and 4 positives), as recorded in models/metrics. Json:

- ROC–AUC: 1.00
- Accuracy: 1.00
- Precision (positive class): 1.00
- Recall (positive class): 1.00

- F1-score (positive class): 1.00



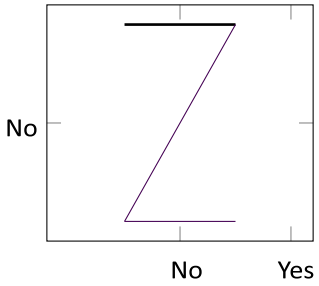


Fig. 2. Bar chart summarizing precision, recall, and F1-score (all equal to 1.0) for the positive class.

These perfect scores reflect the simplicity and small size of the synthetic dataset, and should not be interpreted as real world clinical performance.

VI. DISCUSSION

The proposed system demonstrates that a simple logistic regression model can be effectively integrated into a modern web application to provide fast and interpretable risk estimates. By encapsulating preprocessing and modeling in a single pipeline and exposing the model via a Flask application, the system achieves consistent data handling between training and inference, low-latency predictions suitable for interactive use,

TABLE I
PERFORMANCE METRICS OF THE TRAINED LOGISTIC-REGRESSION MODEL ON THE SYNTHETIC TEST SET.

Metric	Value
ROC–AUC	1.00
Accuracy	1.00
Precision (positive)	1.00
Recall (positive)	1.00
F1-score (positive)	1.00

and a clear separation between machine-learning logic and presentation logic.

The experimental results show perfect performance (ROC– AUC, accuracy, precision, recall, and F1-score all equal to 1.0) on the small synthetic test set. This outcome reflects the limited size and controlled nature of the data rather than real-world difficulty. In practical clinical settings, data are far noisier and more heterogeneous, and such perfect performance should not be expected.

The system also has other limitations: only a limited set of features—primarily lifestyle factors and a few symptoms— are modeled, while other relevant variables such as detailed occupational exposures, comorbidities, family history, and genetic markers are absent. Additionally, the system does not incorporate imaging data, which plays a crucial role in practical lung cancer diagnosis.

For these reasons, the system is best viewed as a teaching and prototyping tool rather than as a clinical decision support system. Nevertheless, it provides a concrete example of how interpretable

statistical models can be deployed in user-friendly interfaces and integrated into wider software ecosystems.

VII. CONCLUSION

This paper has presented a complete, end-to-end implementation of a lung cancer risk prediction system based on logistic regression and deployed as a Flask web application. The system includes synthetic datasets, a reproducible training and evaluation pipeline, and a dual-interface deployment consisting of a web form and a JSON API.

This project demonstrates how core concepts from machine learning and web development can be combined into a single educational application. While the current model and data are not suitable for clinical use, the architecture and codebase provide a practical template for building interpretable, low-latency machine-learning applications in healthcare and other domains.

ACKNOWLEDGMENT

I would like to thank faculty and guide at PDA College of Engineering, Kalaburagi, for their support and guidance during the development of this project.

REFERENCES

[1] American Cancer Society, "Lung Cancer Facts and Figures," [Online]. Available: https://www.cancer.org/. Accessed: 2025.

[2] S. Reddy, J. Fox, and M. P. Purohit, "Artificial intelligence-enabled healthcare delivery," *J. R. Soc. Med.*, vol. 112, no. 1, pp. 22–28, 2019.

[3] F. Pedregosa *et al.*, "Scikit-learn: Machine Learning in Python," *J. Mach. Learn. Res.*, vol. 12, pp. 2825–2830, 2011.

[4] M. C. Tammemagi¨ *et al.*, "Selection criteria for lung-cancer screening," *N. Engl. J. Med.*, vol. 368, no. 8, pp. 728–736, 2013.

[5] P. B. Bach *et al.*, "Benefits and harms of CT screening for lung cancer," *JAMA*, vol. 307, no. 22, pp. 2418–2429, 2012.

[6] A. McWilliams *et al.*, "Probability of cancer in pulmonary nodules detected on first screening CT," *N. Engl. J. Med.*, vol. 369, no. 10, pp. 910–919, 2013.

[7] S. M. Lundberg and S.-I. Lee, "A unified approach to interpreting model predictions," in *Advances in Neural Information Processing Systems*, 2017.

[8] L. Breiman, "Random forests," *Mach. Learn.*, vol. 45, no. 1, pp. 5–32, 2001.

[9] E. J. Topol, "High-performance medicine: the convergence of human and artificial intelligence," *Nat. Med.*, vol. 25, pp. 44–56, 2019.

[10] A. Esteva *et al.*, "A guide to deep learning in healthcare," *Nat. Med.*, vol. 25, pp. 24–29, 2019.

[11] A. Rajkomar, J. Dean, and I. Kohane, "Machine learning in medicine," *N. Engl. J. Med.*, vol. 380, no. 14, pp. 1347–1358, 2019.

[12] A. L. Beam and I. S. Kohane, "Big data and machine learning in health care," *JAMA*, vol. 319, no. 13, pp. 1317–1318, 2018.