Lung Cancer Prediction Web Application Documentation- DSA Project

Mytreyan 2022115102

Project Overview

The Lung Cancer Prediction web application is a comprehensive, data-driven healthcare tool designed to assist clinicians and researchers in predicting lung cancer stages and discovering patient subgroups using machine learning. The system integrates clinical, demographic, and laboratory data to provide real-time predictions, clustering, and rich visual analytics. Built with Flask, scikit-learn, pandas, and modern visualization libraries, the application offers an interactive frontend for data input and result exploration, supporting evidence-based decision-making and research into lung cancer progression.

Objectives

- **Accurate Stage Prediction:** Use patient data to predict lung cancer stage (I–IV) with a logistic regression model, supporting early intervention and personalized treatment.
- **Patient Stratification:** Apply unsupervised clustering (KMeans) to identify subgroups within the patient population, enabling targeted care strategies and research into disease heterogeneity.
- **Interactive Visualization:** Provide clinicians and researchers with dynamic, high-quality data visualizations for exploratory analysis and communication of insights.
- **User-Centric Design:** Offer an intuitive web interface for data entry (sliders, dropdowns, radio buttons) and result interpretation, minimizing barriers to adoption in clinical settings.
- **Scalable and Modular:** Ensure the architecture supports future integration of additional models, features, and data sources.

Problem Statement

Lung cancer remains a leading cause of cancer mortality worldwide. Accurate staging is critical for prognosis, treatment selection, and patient counseling, but manual assessment is time-consuming and prone to subjectivity. Additionally, the diversity of patient presentations and comorbidities complicates risk stratification and resource allocation. There is a pressing need for automated, interpretable tools that can:

- Predict cancer stage from multidimensional data.
- Reveal clinically relevant patient clusters for tailored management.
- Present complex data in accessible, actionable formats for healthcare professionals.

Dataset Description

Source: lung_cancer_data.csv
Rows: 2,000+ patients
Columns/Features: 40+ variables per patient

Feature Categories & Data Types:

Feature Name	Description	Data Type	Example Values
Patient_ID	Unique identifier	object	Patient21982
Age	Age in years	int	31, 43
Gender	Biological sex	object	Male, Female
Smoking_History	Smoking status	object	Never Smoked, Former Smoker, Current Smoker
Tumor_Size_mm	Tumor size (mm)	float	40.2, 32.1
Tumor_Location	Anatomical lobe	object	Upper Lobe, Middle Lobe
Stage	Cancer stage (target)	object	Stage I, Stage II
Treatment	Primary treatment type	object	Surgery, Chemotherapy
Survival_Months	Survival duration	int	97, 109
Ethnicity	Patient ethnicity	object	Asian, Hispanic

Feature Name	Description	Data Type	Example Values
Insurance_Type	Insurance coverage	object	Private, Medicaid
Family_History	Family history of cancer	object	Yes, No
Comorbidity_*	Comorbid conditions (multiple columns)	object	Yes, No
Performance_Status	Functional status	int	0–4
Blood_Pressure_Systolic	Systolic BP	int	120, 140
Blood_Pressure_Diastolic	Diastolic BP	int	80, 90
Blood_Pressure_Pulse	Pulse pressure	int	70, 75
Hemoglobin_Level	Hemoglobin (g/dL)	float	13.2, 15.4
White_Blood_Cell_Count	WBC count (10^9/L)	float	8.1, 9.7
Platelet_Count	Platelets (10^9/L)	float	250, 300
Albumin_Level	Serum albumin (g/dL)	float	4.0, 3.5
...	Additional laboratory and clinical values	float/int	
Smoking_Pack_Years	Smoking exposure	float	10.5, 35.0

Target Variable:

- Stage (categorical, ordinal: I–IV)

Data Preprocessing

1. Label Encoding

- All categorical variables (e.g., Gender, Smoking_History, Tumor_Location, Treatment, Ethnicity, Insurance_Type, Family_History, Comorbidities) are converted to numeric codes using LabelEncoder.
- Special handling for binary features (Yes/No → 1/0).
- Encoders are stored for consistent transformation of user input and decoding predictions.

2. Feature Scaling

- All numerical features are standardized with StandardScaler (zero mean, unit variance) to ensure balanced model training and effective clustering.

3. Missing Value Handling

- The dataset appears complete, but in production, missing values would be imputed (mean for continuous, mode for categorical).

4. Train-Test Split

- Data is split into training and testing sets (80/20) with stratification on the Stage variable to preserve class distribution.

Machine Learning Algorithms

Logistic Regression for Stage Prediction

- **Purpose:** Multiclass classification to predict cancer stage (I–IV).
- **Features:** All relevant clinical, demographic, and laboratory variables, excluding identifiers and target.
- **Training:** Model is fit on scaled training data. Hyperparameters (e.g., max_iter=1000) ensure convergence.
- **Prediction:** User input is preprocessed identically, scaled, and passed to the model for stage prediction.
- **Interpretability:** Coefficients can be analyzed for feature importance.

KMeans Clustering for Patient Grouping

- **Purpose:** Unsupervised learning to discover latent patient clusters based on clinical/lab features.
- **n_clusters:** 3 (empirically chosen; can be tuned).
- **Features:** Same as above, excluding target.
- **Output:** Cluster labels are assigned to each patient; used for visualization and potential stratified analysis.

Model Training, Evaluation, and Prediction

Training

- Logistic Regression is trained on the scaled training set.
- KMeans is fit on the entire (processed and scaled) dataset for clustering.

Evaluation

- Logistic Regression: Evaluated on the test set using classification metrics (accuracy, precision, recall, F1-score). The classification_report from scikit-learn provides detailed breakdowns.
- KMeans: Evaluated qualitatively via visualization (e.g., cluster separation in 2D/3D plots) and silhouette scores.

Prediction

- User input is collected via the web form, encoded/scaled as per training, and passed to the model for prediction.
- Predicted stage is decoded back to human-readable labels for display.

Web Application Architecture

Backend (Flask)

- **Structure:**
    - app.py contains route definitions, data loading, preprocessing, model training, and prediction logic.
    - Data and models are loaded once at startup for efficiency.
- **Key Routes:**

    - /predict: Main prediction page. Handles GET (form display) and POST (prediction) requests. Dynamically generates forms based on feature metadata.
    - /clusters: Displays cluster analysis page.
    - /clusters/image: Returns a cluster plot image (2D scatter with cluster coloring).
    - /visualize: Lists available visualizations.
    - /visualize/plot/<plot_name>: Generates and returns requested plot images (histograms, boxplots, 3D scatter, heatmaps, etc.).

Frontend Design

- **Dynamic Forms:**
    - Numerical features: Rendered as sliders (min/max based on data).
    - Categorical features: Rendered as dropdowns.
    - Binary features (Yes/No): Rendered as radio buttons.
    - All forms are generated dynamically based on the features present in the dataset.
- **Result Display:**
    - Predicted stage is shown on the prediction page.
    - Cluster assignment and visualizations are displayed as images or tables.

Visualization Techniques

Matplotlib and Seaborn are used to generate a variety of static and interactive plots, including:

- **Histograms:** Age distribution, tumor size, etc.
- **Boxplots:** Tumor size by stage, survival months by stage.
- **3D Scatter Plots:**
    - Age vs. Tumor Size vs. Survival Months (colored by stage or cluster).
    - Blood Pressure Systolic vs. Diastolic vs. Hemoglobin.
- **Heatmaps:** Correlation matrices to reveal relationships between features.
- **Violin Plots:** Survival months by stage.
- **Count Plots:** Gender, smoking history, treatment type.
- **Pairplots:** Multivariate relationships among numeric features.

Technology Stack

Layer	Tools/Libraries
Programming	Python 3.x
Web Framework	Flask
Data Handling	pandas, numpy
Machine Learning	scikit-learn
Visualization	matplotlib, seaborn
HTML/CSS	Jinja2 templates, Bootstrap (for styling)
Deployment	Gunicorn, Docker (optional)

Benefits and Potential Applications in Healthcare

- **Clinical Decision Support:** Provides instant, data-driven stage predictions, reducing diagnostic delays and supporting evidence-based treatment planning.
- **Population Health Management:** Clustering reveals patient subgroups, informing targeted screening, resource allocation, and personalized intervention strategies.
- **Research Enablement:** Rich visualizations and clustering facilitate hypothesis generation, cohort discovery, and outcome analysis for researchers.
- **Patient Engagement:** The intuitive interface can be adapted for patient-facing risk assessments and education.
- **Scalability:** The modular design allows for integration of new data sources (e.g., genomics, imaging) and extension to other cancers or diseases.

In summary, this project exemplifies the practical integration of machine learning and web technologies to address real-world challenges in lung cancer care, offering a robust foundation for further innovation in digital health.