Lung Cancer Prediction using Big Data Technologies

Enhancing Accuracy with Apache Spark and Hadoop

Yangoti Sai Bhavya 122ad0007

Dumpala Rohith 122ad0013

March 31, 2025

Overview

Introduction

- Existing research used traditional ML models like for lung cancer prediction.
- No parallel processing tools like Apache Spark were utilized.
- Computational inefficiencies and scalability limitations.
- Accuracy could be further improved.

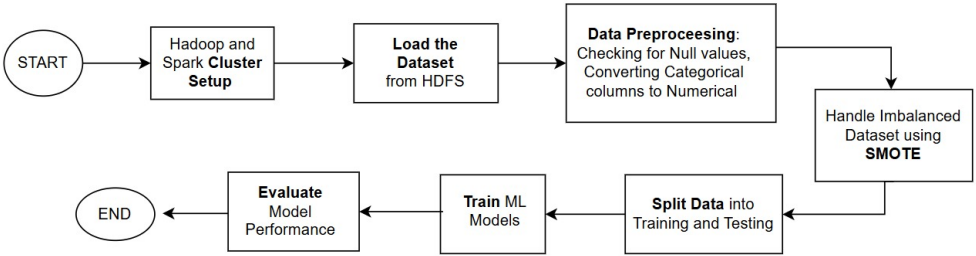
Objectives

- Use Big Data Approaches like Spark to improve predictive accuracy.
- Leverage parallel processing for efficient computations.
- Compare results with the existing research work.

Technologies Used

- **Apache Hadoop** - Distributed storage using HDFS (Hadoop Distributed File System) for handling large-scale medical data.
- **Apache Spark** - High-speed data analytics with in-memory computation and MLlib for machine learning.
- **Spark Cluster Setup** - Single-node cluster with Master and Worker nodes for efficient parallel processing.
- **Jupyter Notebook** - Used for interactive data analysis and model visualization.
- **Machine Learning Models** - Implemented various ML algorithms to improve accuracy.

Methodology



Cluster Setup

- **Step 1:** Installed Apache Hadoop for distributed storage using HDFS.
- **Step 2:** Installed Apache Spark for high-speed data processing and machine learning.
- **Step 3:** Set up a single-node cluster, with a Master node and Worker node, on a local machine.
- **Step 4:** Configured Spark to run on localhost for parallel processing.
- **Step 5:** Connected Jupyter Notebook to Spark for interactive data analysis and machine learning tasks.

Dataset Description

The number of attributes: 16
The number of instances: 310
Attributes of the dataset:

- Gender: M (male), F (female)
- Age: Age of the patient
- Smoking: YES=2, NO=1
- Yellow fingers: YES=2, NO=1
- Anxiety: YES=2, NO=1
- Peer-pressure: YES=2, NO=1
- Chronic Disease: YES=2, NO=1
- Fatigue: YES=2, NO=1
- Allergy: YES=2, NO=1
- Wheezing: YES=2, NO=1
- Alcohol: YES=2, NO=1
- Coughing: YES=2, NO=1
- Shortness of Breath: YES=2, NO=1
- Swallowing Difficulty: YES=2, NO=1
- Chest pain: YES=2, NO=1
- Lung Cancer: YES, NO

Data Preprocessing

Steps involved:

- Checking for missing values
- Converting categorical columns to numerical values

We have two categorical columns, **GENDER** and **LUNGCANCER**, that need to be converted into numerical form for easy processing.

Example of Converted Data:

- GENDER: M=0, F=1
- LUNGCANCER: YES=0, NO=1

Data Balancing and Train-Test Split

Handling Imbalanced Data: SMOTE

- The dataset was highly imbalanced, leading to biased predictions.
- Used **SMOTE (Synthetic Minority Over-sampling Technique)** to generate synthetic samples for the minority class.
- SMOTE creates synthetic data points by interpolating between existing minority class instances.
- This helps in preventing overfitting while maintaining meaningful data distribution.

Train-Test Split

- The dataset was divided into **80% training** and **20% testing**.
- Ensured that the model was evaluated on unseen data for reliable performance measurement.

ML Models and Their Performance

Logistic Regression

- A simple yet effective classification algorithm used for binary classification.
- Computes the probability of lung cancer presence using a sigmoid function.
- Achieved an accuracy of 94%.

	precision	recall	f1-score	support
0.0	0.92	0.96	0.94	50
1.0	0.96	0.92	0.94	50
accuracy			0.94	100
macro avg	0.94	0.94	0.94	100
weighted avg	0.94	0.94	0.94	100

Random Forest

- An ensemble learning method combining multiple decision trees for better accuracy.
- Reduces overfitting compared to a single decision tree by averaging predictions.
- Achieved an accuracy of 96%.

	precision	recall	f1-score	support
0.0	0.94	0.98	0.96	50
1.0	0.98	0.94	0.96	50
accuracy			0.96	100
macro avg	0.96	0.96	0.96	100
weighted avg	0.96	0.96	0.96	100

K-Nearest Neighbors (KNN)

- A distance-based algorithm that classifies data points based on their nearest neighbors.
- Works well with small datasets but may struggle with high-dimensional data.
- Achieved an accuracy of 95%.

	precision	recall	f1-score	support
0.0	0.94	0.96	0.95	50
1.0	0.96	0.94	0.95	50
accuracy			0.95	100
macro avg	0.95	0.95	0.95	100
weighted avg	0.95	0.95	0.95	100

Decision Tree

- A tree-structured model where decisions are made based on feature splits.
- Prone to overfitting, but effective for interpretability.
- Achieved an accuracy of 94%.

	precision	recall	f1-score	support
0.0	0.92	0.96	0.94	50
1.0	0.96	0.92	0.94	50
accuracy			0.94	100
macro avg	0.94	0.94	0.94	100
weighted avg	0.94	0.94	0.94	100

Gradient Boosting

- A boosting algorithm that builds models sequentially, correcting errors at each step.
- Works well with structured data but can be computationally expensive.
- Achieved an accuracy of 95%.

	precision	recall	f1-score	support
0.0	0.94	0.96	0.95	50
1.0	0.96	0.94	0.95	50
accuracy			0.95	100
macro avg	0.95	0.95	0.95	100
weighted avg	0.95	0.95	0.95	100

Linear Regression

- Typically used for continuous target variables but tested here for classification.
- Predicts a continuous score that is thresholded to classify lung cancer presence.
- Achieved an accuracy of 93%.

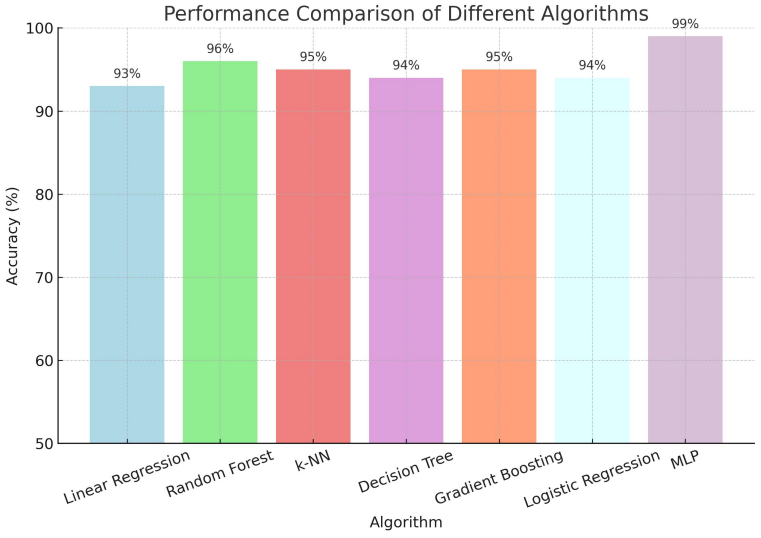
	precision	recall	f1-score	support
0.0	0.94	0.92	0.93	50
1.0	0.92	0.94	0.93	50
accuracy			0.93	100
macro avg	0.93	0.93	0.93	100
weighted avg	0.93	0.93	0.93	100

Multi-Layer Perceptron (MLP) - Proposed Technique

- A neural network-based approach for capturing complex patterns.
- Requires hyperparameter tuning for optimal performance.
- Achieved an accuracy of 99%.

	precision	recall	f1-score	support
0.0	1.00	0.98	0.99	50
1.0	0.98	1.00	0.99	50
accuracy			0.99	100
macro avg	0.99	0.99	0.99	100
weighted avg	0.99	0.99	0.99	100

Performance Comparison Graph

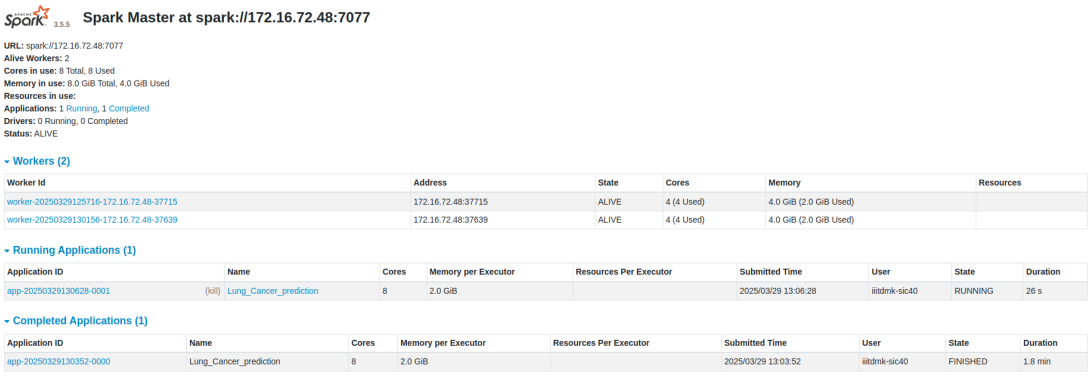


Results and Comparison

Algorithm	Efficiency
Linear regression	58.3653
Random forest	95.3703
k-nearest neighbor	90.7407
Decision Tree	94.4444
Gradient Boosting	94.4444

Algorithm	Accuracy
Linear Regression	93%
Random Forest	96%
k-Nearest Neighbor	95%
Decision Tree	94%
Gradient Boosting	95%
Additional Algorithms	
Logistic Regression	94%
Multi-Layer Perceptron (MLP)	99%

Single Node Setup - Master



Spark 3.5.5 **Spark Master at spark://172.16.72.48:7077**

URL: spark://172.16.72.48:7077
Alive Workers: 2
Cores in use: 8 Total, 8 Used
Memory in use: 8.0 GiB Total, 4.0 GiB Used
Resources in use:
Applications: 1 Running, 1 Completed
Drivers: 0 Running, 0 Completed
Status: ALIVE

Workers (2)

Worker Id	Address	State	Cores	Memory	Resources
worker-20250329125716-172.16.72.48-37715	172.16.72.48:37715	ALIVE	4 (4 Used)	4.0 GiB (2.0 GiB Used)	
worker-20250329130156-172.16.72.48-37639	172.16.72.48:37639	ALIVE	4 (4 Used)	4.0 GiB (2.0 GiB Used)	

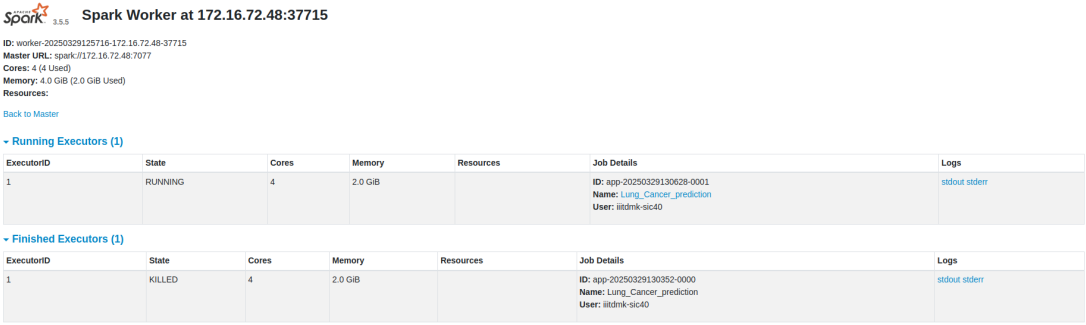
Running Applications (1)

Application ID		Name	Cores	Memory per Executor	Resources Per Executor	Submitted Time	User	State	Duration
app-20250329130628-0001	(kill)	Lung_Cancer_prediction	8	2.0 GiB		2025/03/29 13:06:28	iiitdmk-sic40	RUNNING	26 s

Completed Applications (1)

Application ID	Name	Cores	Memory per Executor	Resources Per Executor	Submitted Time	User	State	Duration
app-20250329130352-0000	Lung_Cancer_prediction	8	2.0 GiB		2025/03/29 13:03:52	iiitdmk-sic40	FINISHED	1.8 min

Single Node Setup - Worker



Spark 3.5.5

Spark Worker at 172.16.72.48:37715

ID: worker-20250329125716-172.16.72.48-37715
Master URL: spark://172.16.72.48:7077
Cores: 4 (4 Used)
Memory: 4.0 GiB (2.0 GiB Used)
Resources:

Back to Master

Running Executors (1)

ExecutorID	State	Cores	Memory	Resources	Job Details	Logs
1	RUNNING	4	2.0 GiB		**ID:** app-20250329130628-0001 **Name:** Lung_Cancer_prediction **User:** iiitdmk-sic40	stdout stderr

Finished Executors (1)

ExecutorID	State	Cores	Memory	Resources	Job Details	Logs
1	KILLED	4	2.0 GiB		**ID:** app-20250329130352-0000 **Name:** Lung_Cancer_prediction **User:** iiitdmk-sic40	stdout stderr

Conclusion and Future Work

- By incorporating Multi-Layer Perceptron (MLP), we significantly improved accuracy compared to traditional methods.
- The use of Apache Spark reduced computation time and enhanced scalability.
- Parallel processing played a crucial role in handling large datasets efficiently.
- **Future Work:**
  - Extending the approach to process image data using deep learning models.
  - Leveraging distributed computing frameworks to handle even larger datasets.
  - Exploring advanced optimization techniques for further performance improvements.

Thank You!