Final Report: Lung Cancer Detection

Cristian Degni, Enrico Sbuttoni

21/01/2025

Abstract

Our project introduces a potential solution for predicting lung cancer by analyzing patient data and behavior. This report delves into the project's motivation, methodologies, and results, offering a thorough assessment of its strengths and areas for improvement. The primary goal is to develop a reliable system for early detection of lung cancer.

To achieve this, we employed feature engineering, supervised learning, dataset manipulation techniques, and ensemble methods. The implementation integrates algorithms like SVM, Random Forest, and XGBoost while incorporating techniques such as SMOTE and PCA to enhance performance.

We evaluated the model using standard metrics, such as precision, recall, accuracy, and F1-score.

Introduction

The project addresses the challenge of predicting lung cancer by analyzing patient data and behavior. This problem holds significant importance due to its potential to enable early diagnosis, which is crucial for improving survival rates and reducing the severity of treatments. The project's aim is to develop a reliable, simple, data-driven system for early detection of lung cancer using only data easily available from the patient's medical file.

Inputs and Outputs

The input for the algorithm includes structured datasets consisting of attributes such as *gender, age, smoking, yellow fingers, anxiety, peer pressure, chronic disease, fatigue, allergy, wheezing, alcohol consuming, coughing, shortness of breath, swallowing difficulty, and chest pain.* The output consists of a binary classification indicating whether the patient is likely to have lung cancer (YES or NO). The methodology involves testing *"raw"* machine learning model and then try to achieve a performance improvement through feature engineering and dataset manipulation.

Related Work

We used several references as an inspiration for our work, the principal was:

- **[https://www.kaggle.com/code/hasibalmuzdadid/lung-cancer-analysis-accuracy-96-4]:** Explores several ML techniques and achieves great outcome. However, it lacks data preprocessing and improvements.

The project's unique contribution lies in its slightly higher results and in the fact that we tried to expand the model we discovered to other dataset, determining it is possible to create a reliable model.

Dataset and Features

Dataset Description

The project utilizes two datasets (from now on Dataset 1 and Dataset 2) with distinct characteristics for training, validation, and testing but the same schema. Key details are as follows:

- **Dataset 1:**

  - **Structure:** Tabular, CSV format
  - **Size:** 308 records
  - **Source:** Verified and high-quality data
  - **Characteristics:** Easy to classify due to clean and consistent structure

- **Dataset 2:**
  - **Structure:** Tabular, CSV format, same schema as Dataset 1
  - **Size:** 3000 rows
  - **Source:** Potentially artificially generated
  - **Characteristics:** Larger and noisier, potentially requiring additional preprocessing

Three distinct analyses were performed:

1. **Analysis 1:** Using only Dataset 1 (308 records).
2. **Analysis 2:** Combining Dataset 1 with a random subset of Dataset 2 (308 + 1200 rows from Dataset 2, randomly sampled).
3. **Analysis 3:** Combining Dataset 1 with a subset of Dataset 2 (308 + 1200 rows) where entries from Dataset 2 were selected to match the label distribution of Dataset 1.

These analyses were designed to explore whether the value of Dataset 1 could be leveraged to classify parts of Dataset 2. We did not use the entire Dataset 2 because preliminary analyses revealed limited utility in the data, indicating potential issues with its overall quality or informativeness.

0.1 Feature Distribution and Correlation

Generally speaking the label distribution for Analysis 1 and 3 is 83% YES and 17% NO while for Analysis 2 the distribution is 67% YES and 33% NO.

Analysis 1

The feature correlations in Analysis 1 indicate that the strongest negative correlation with the target variable is observed for **ALLERGY** (-0.333), **ALCOHOL CONSUMING** (-0.294), and **SWALLOWING DIFFICULTY** (-0.265). These features could potentially serve as important indicators for classification. The weaker correlations, such as **SHORTNESS OF BREATH** (-0.054), suggest that these features may not have a strong direct relationship with the target.

Analysis 2

In Analysis 2, feature correlations appear less pronounced. Notably, **PEER_PRESSURE** shows a slight positive correlation (0.013), while most other features exhibit weak negative correlations, such as **ALLERGY** (-0.131) and **ALCOHOL CONSUMING** (-0.124). This suggests that the random subset from Dataset 2 introduces additional noise, making it harder to establish strong relationships with the target.

Analysis 3

In Analysis 3, correlations remain relatively weak, with **ALLERGY** (-0.108) and **COUGHING** (-0.075) continuing to show moderate negative relationships. The closer alignment of label distributions between the two datasets does not seem to strengthen the feature-target correlations significantly, indicating that matching label distributions alone may not enhance feature utility.

Preprocessing

Steps include:

1. **Initial Attempts:** Experiments were conducted without preprocessing to establish a baseline.

2. **Balancing the Dataset:** Applied Synthetic Minority Oversampling Technique (SMOTE) to balance the dataset.

3. **Dimensionality Reduction:** Principal Component Analysis (PCA) was used to reduce dimensionality while retaining significant variance. After each preprocessing step, data was re-split into training and testing sets. The explained variance ratios for the PCA were as follows:

   - Analysis 1: [0.11121844, 0.08551355]
   - Analysis 2: [0.11697006, 0.09510812]
   - Analysis 3: [0.11697006, 0.09510812]

   These values indicate that the first two principal components captured a moderate amount of variance, sufficient for dimensionality reduction without significant information loss.

4. **Feature Engineering:** Created new interaction features:

   - `Age*Smoking`: Interaction between AGE and SMOKING.
   - `Chronic*Anxiety`: Interaction between CHRONIC DISEASE and ANXIETY.

   After creating new features, the dataset was split into training and testing sets, standardized using `StandardScaler` and then used for the training of the models.

Methods

Algorithmic Framework

The project's backbone involves a comparison of **Random Forest**, **Support Vector Machine (SVM)** and **XGBoost** classifiers to evaluate their performance on given data. It follows the principles of **supervised learning**, where models are trained on labeled data to predict class labels. The techniques employed include **ensemble learning** (Random Forest and XGBoost) and **kernel-based learning** (SVM), allowing for a robust and flexible approach to classification tasks.

Key Steps

1. **Model Design:** Three different models were evaluated for classification:

   - **Random Forest**: An ensemble learning method that constructs multiple decision trees and aggregates their predictions to improve accuracy and reduce overfitting.
   - **SVM**: A classification algorithm that finds the optimal hyperplane to separate classes, evaluated with different kernels (linear, RBF, polynomial) to identify the best-performing configuration.
   - **XGBoost**: A gradient boosting algorithm that builds an ensemble of decision trees sequentially, optimizing performance through regularization and efficient computation.

2. **Training:** Model training incorporated systematic hyperparameter tuning:

   - **Random Forest**: Tuned hyperparameters included `n_estimators` (10, 50, 100, 200), `max_depth` (None, 10, 20, 30), `min_samples_split` (2, 5, 10), `min_samples_leaf` (1, 2, 4), and `bootstrap` (True, False).
   - **SVM**: Tuned hyperparameters included `C` (0.1, 1, 10, 100), `kernel` (linear, RBF), `gamma` (scale, auto), and `class_weight` (None, balanced).
   - **XGBoost**: Tuned hyperparameters included `max_depth` (1, 10, 100), `n_estimators` (50, 100), `learning_rate` (0.01, 0.1, 0.3), and `scale_pos_weight` (1, 2, 3).

3. **Validation:** The models were evaluated using **10-fold cross-validation** for all models, ensuring robustness by partitioning the dataset into training and validation subsets. **GridSearchCV** was employed for hyperparameter tuning with a greedy approach. Performance metrics included accuracy, classification reports, and confusion matrices to analyze misclassifications.

Experiments, Results, and Discussion

Experimental Setup

The repository was evaluated using:

- **Metrics:** Precision, Recall, F1-score, and Accuracy.

Results and Discussion

We tried 4 measurements for each model (simple, with SMOTE, after tuning the hyper-parameters and with preprocessing). Therefore we got 38 distinct results. In the following subsections we will explore the best result for each analysis and model while highlighting some particular outcomes of other measurement.

Analysis 1

The results for Analysis 1 showed that the model performed well on Dataset 1, achieving high precision, recall, and F1-score due to the clean and consistent nature of the data. It is still favored class YES due to the distribution of the data.

Metric	**NO**	**YES**	**Overall**
Accuracy (%)	-	-	92.00
Precision (%)	71.00	95.00	92.00
Recall (%)	62.00	96.00	92.00
F1-Score (%)	67.00	95.00	92.00
Support	8	55	63

Table 1: Performance Metrics of the ***Random Forest Model*** with SMOTE/Cross-Validation on the Test Set

Metric	**NO**	**YES**	**Overall**
Accuracy (%)	-	-	94.00
Precision (%)	75.00	96.00	94.00
Recall (%)	75.00	96.00	94.00
F1-Score (%)	75.00	96.00	94.00
Support	8	55	63

Table 2: Performance Metrics of the ***SVM Model*** with SMOTE on the Test Set

Metric	**NO**	**YES**	**Overall**
Accuracy (%)	-	-	92.00
Precision (%)	64.00	98.00	94.00
Recall (%)	88.00	93.00	92.00
F1-Score (%)	74.00	95.00	93.00
Support	8	55	63

Table 3: Performance Metrics of the ***XGBoost Model*** with SMOTE on the Test Set

The best model turned out to be the SVM with SMOTE preprocessing while being also the most balanced model with a weighted average F1-Score of 94.00.

Analysis 2

In Analysis 2, the inclusion of random samples from Dataset 2 introduced noise, leading to a slight decline in performance metrics compared to Analysis 1. However, the model still demonstrated acceptable accuracy and recall.

Metric	**NO**	**YES**	**Overall**
Accuracy (%)	-	-	69.00
Precision (%)	52.00	74.00	67.00
Recall (%)	33.00	86.00	69.00
F1-Score (%)	40.00	79.00	67.00
Support	94	206	300

Table 4: Performance Metrics of the ***Random Forest Model*** with Feature Engineering on the Test Set

Metric	**NO**	**YES**	**Overall**
Accuracy (%)	-	-	67.00
Precision (%)	48.00	76.00	67.00
Recall (%)	47.00	77.00	67.00
F1-Score (%)	47.00	76.00	67.00
Support	94	206	300

Table 5: Performance Metrics of the ***SVM Model*** with Cross-Validation on the Test Set

Metric	**NO**	**YES**	**Overall**
Accuracy (%)	-	-	67.00
Precision (%)	53.00	70.00	64.00
Recall (%)	29.00	86.00	67.00
F1-Score (%)	38.00	77.00	64.00
Support	103	197	300

Table 6: Performance Metrics of the ***XGBoost Model*** with Feature Engineering on the Test Set

In analysis 2 we got the most interesting results and opportunity to grow in our opinion because the dataset has a different distribution but can achieve some decent result. The feature engineering techniques have allowed us to achieve improvements and in fact the model with better accuracy is the Random Forest after adding features; the most balanced model, on the other hand, is the SVM, which can to recognise even NO class samples more consinstently. Some notable results:

- In XGBoost without any modification we get 63% accuracy while after applying Cross-Validation and feature engineering we improve to 67%.
- The SVM model without enhancements on the other hand has a lower accuracy than the post pre-processing model but manages to have higher recall and precision for the NO class.

Analysis 3

Analysis 3 yielded metrics similar to Analysis 1. This shows that if the dataset increases but we maintain a substantial consistency of distribution of the labels, the model retains much of its predictive power.

Metric	**NO**	**YES**	**Overall**
Accuracy (%)	-	-	87.00
Precision (%)	100.00	86.00	88.00
Recall (%)	9.00	100.00	87.00
F1-Score (%)	17.00	93.00	82.00
Support	44	256	300

Table 7: Performance Metrics of the ***Random Forest Model*** with Feature Engineering on the Test Set

Metric	**NO**	**YES**	**Overall**
Accuracy (%)	-	-	0.81
Precision (%)	0.33	0.0.88	0.80
Recall (%)	0.30	0.89	0.81
F1-Score (%)	0.31	0.89	0.80
Support	44	256	300

Table 8: Performance Metrics of the ***SVM Model*** with Cross-Validation on the Test Set

Metric	**NO**	**YES**	**Overall**
Accuracy (%)	-	-	86.00
Precision (%)	23.00	89.00	82.00
Recall (%)	9.00	96.00	86.00
F1-Score (%)	13.00	93.00	84.00
Support	34	266	300

Table 9: Performance Metrics of the ***XGBoost Model*** with SMOTE on the Test Set

It can be seen that most of the accuracy values are greatly improved compared to the previous dataset despite having the same number of samples. This was at the expense of predictive ability on the NO because this dataset is very unbalanced and struggles to find correlations within the data to correctly predict both classes, preferring to go for the one that is statistically more likely. The three types of models behave almost equally in terms of both accuracy and Recall while Random Forest has a high precision value for the NO class: this means that the classified samples must have strong evidence of belonging to that class and in that case the model is very precise.

Some notable results:

- In the SVM model at the beginning, i.e., without preprocessing and feature engineering, the model is much more uncertain with an accuracy of only 58% while after the various steps there is always an improvement with the consequent loss of predictive power on the NO.

Discussion

- **Strengths:**
  - Achieved high accuracy on the small and balanced dataset.
  - Demonstrated the ability to transfer value from the real dataset to the artificial dataset, improving its utility.
- **Limitations:**
  - Low recall for the ”NO” class, which, while acceptable given the preference for false positives over false negatives in a medical application, still represents a limitation.
  - Degradation of model performance when the size of the artificial dataset is increased significantly, highlighting the importance of dataset quality.

Conclusion and Future Work

Our Project successfully explored various predictive models to forecast lung cancer using only patient habits and clinical history. This methodology highlights the potential of machine learning as a first-step screening tool. While not fully reliable for definitive diagnoses, this approach aims to exclude individuals who show no apparent risk of lung cancer, thus streamlining the focus on higher-risk cases.

To ensure robustness, we utilized three different datasets to observe how model performance changes across varying data distributions. Despite the inherent challenges posed by imbalanced datasets, we are satisfied with our results. The project demonstrates how predictive modeling can contribute significantly to early-stage health assessments and efficient allocation of medical resources.

Future Directions

Potential improvements include:

1. **Dataset Expansion:** Incorporating lung X-ray images instead of clinical history. While this could improve robustness, it would significantly increase computational requirements and necessitate the use of neural networks, moving away from the simplicity of the current approach.
2. **Algorithm Optimization:** Implementing deeper preprocessing and feature engineering steps to better capture relationships within the data.
3. **Model Exploration:** Experimenting with advanced models, such as neural networks, to potentially enhance predictive performance.

Contributions

- **Cristian Degni:** Focused on the implementation of the code, led the analysis of results, and contributed to the shared report writing.
- **Enrico Sbuttoni:** Concentrated on dataset exploration, managed preprocessing tasks, and contributed significantly to the shared report writing, particularly in non-analytical sections.

References

1. Kumar et al., "Machine Learning for Early Detection of Lung Cancer," IEEE Transactions on Biomedical Engineering, 2021.
2. Pedregosa et al., "Scikit-learn: Machine Learning in Python," Journal of Machine Learning Research, 2011.
3. "XGBoost Documentation: Scalable and Flexible Gradient Boosting," available at `https://xgboost.ai`.
4. ChatGPT and other AI tools for both the report and the code implementation. Especially useful to correct code errors and to rephrase in a better way our analysis.

Failures

Initially, our goal was to predict the number of months of life remaining for patients, leveraging clinical and historical data. Unfortunately, this idea had to be abandoned because the dataset used did not demonstrate any predictive value. This failure underscores the importance of dataset quality and relevance in machine learning projects. While disappointing, this experience guided us to pivot toward a more achievable and impactful goal of using predictive modeling as a first-step screening tool for lung cancer.