Lung Cancer Detection using Machine Learning: An Evaluation of Deep Learning Models

Nolan Karsok
Khoury College of Computer Sciences
Northeastern University

Abstract

Lung cancer is a leading cause of death globally, and early detection is crucial to increase the chances of successful diagnosis and treatment. Machine learning models have been proposed to predict lung cancer using clinical, imaging, and genetic data. While being highly sophisticated in diagnostics, these clinical and imaging data algorithms can be expensive to collect and build. However, lung cancer differs from other types in the fact that often times, symptoms of the disease are tangible and physical, such as coughing, shortness of breathe and chest pain. While not conclusive in and of itself, an individual that reports any combination of a variety of side effects may be at a greater risk of a diagnosis of lung cancer. A project like this can be used in medical practice by creating a more cost effective way to diagnosis a high risk of lung cancer.

I. Introduction

The goal of this project is to build a machine learning model that can accurately predict the presence of lung cancer in patients. Various clinical variables, including age, gender, smoking status, presence of anxiety, and coughing are the features used to train a binary classification model. The model will predict whether a patient has lung cancer or not based on the input variables. The aim is to build a model with high accuracy that can assist physicians in making clinical decisions and identifying high-risk patients and also give individuals the presence of mind to admit themselves to medical treatment if they believe they are at risk. The dataset is a publicly available dataset and contains 15 features of survey data around these clinical attributes and is a public dataset from the online lung cancer detection system.

II. Data Preprocessing

A series of data preprocessing techniques was needed to effectively build a predictive machine learning algorithm on this dataset. The raw data was heavily skewed towards positive lung cancer cases (87% labeled as having lung cancer). An imbalanced dataset such as this will often lead to biased models and overfitting. In order to account for this imbalance, the minority class was resampled to even the distribution. The dataset was originally labeled with 1's and 2's as opposed to traditional binary values 0 and 1, so one hot encoding was performed to allow for ease of analysis and modeling. Finally, a 70/30% train test split was used for this analysis.

III. Feature Selection

Feature selection is a necessary step in the modeling process in order to properly fit the algorithm to the data and avoid overfitting. With a dataset with 15 features such as this one, there is a significant risk of overfitting an algorithm if all 15 features were used.The first method used for feature selection was a correlation matrix. This is a simple and effective tool to understand if there are any two features that are highly correlated with eachother leading to the risk of overfitting. The correlation matrix for this dataset did not show any correlations that were strong enough to remove them at this stage of feature selection.

To narrow down the list of features to be modeled, the selectKbest method was used with the chi-squared to the function. The final result was a selection of ten features to be used in modeling: Age, Yellow Fingers, Anxiety, Peer Pressure, Allergy, Wheezing, Alcohol Consumption, Coughing, Swallowing Difficulty, Chest Pain.

IV. Implementation

A. *Logistic Regression*

As a binary classification algorithm, logistic regression aims to determine if a given observation belongs to a specific class or not. The model employs a logistic function to map input values to probabilities between 0 and 1, with the output

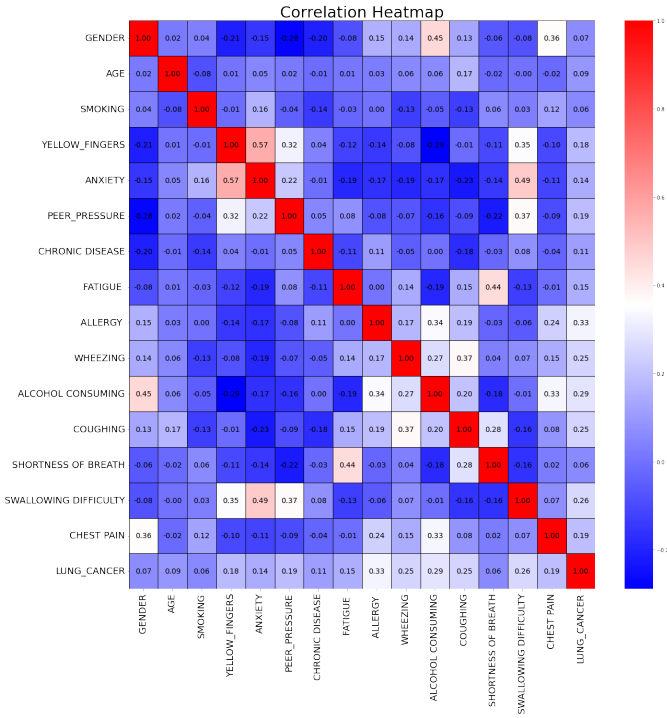

Figure 1: Correlation Matrix of all features.

reflecting the likelihood of the observation belonging to the positive class, and trained through maximum likelihood estimation. Logistic regression offers a simplistic implementation, however degrades in performance if the data is not linear.

B. *K-Nearest Neighbors*

K-Nearest Neighbors (KNN) is a non-parametric, supervised model that can be used for both regression and classification problems. The model works by finding the k closest neighbors to a given observation in the training data and then estimating the likelihood that the test data will become a member of a specific class. Tuning the parameter k will yield an increased performance for this algorithm.

C. *Random Forest*

Random Forest is an ensemble model that combines multiple decision trees to improve performance. Each tree in the ensemble is trained and provides a prediction on the test data. Then, the model's final prediction is the class with the most votes coming from all the trees in the forest. Random forest models often provide a strong performance on test data and generally will outperform a single decision tree model, however will take more time and resources to be able to properly train given the complexity.

D. *Gradient Boosting*

Another ensemble model, gradient boosting, uses weaker models to give a final prediction. The weaker models, typically being decision trees, are combined with the goal of each supplemental model addition being used to correct the weaknesses of the prior model.

E. *Support Vector Machine*

Support Vector Machine (SVM) is a model that finds the optimal hyperplane that separates the data into different classes. The model looks to maximize the distance between data points in each binary class, which allows the algorithm to distinctly predict which set the output belongs to. These hyperplanes are called decision boundaries.

F. *Decision Tree*

Decision tree is a model that works by splitting the features and creating a tree-like structure with each node representing a decision that helps to classify the target variable. This is a simple model which will often lead to overfitting on high-dimensional data.

V. Results

A. *Logistic Regression*

The logistic regression algorithm performed moderately well on this dataset, with an accuracy score of 88%. The model was one of the most consistent across positives and negatives cases of lung cancer, where the Type I error rate was 15% and Type II error was 12%. While this consistency is an indicator that the algorithm will scale well on new test data and not overfit, the Type II error will be more essential to minimize as false negatives can be a serious issue in medical diagnosis as they can lead to delayed or missed treatment, which can have negative impacts on patient outcomes. Being a simple and effective model to run, this logistic regression model would be reasonable to put into practice, however other algorithms may yield better accuracy and recall for the positive cases of lung cancer (false negative rate).

B. *K-Nearest Neighbors*

The first step in building the KNN algorithm was to utilize hyperparameter tuning to find the optimal value of k. A value of k too low will result in a model that does not handle variance well whereas a value too large will result in underfitting. To find this optimal value, KNN models were trained and looped through each value of k and recorded the highest accuracy score. For the final model, the optimal value of k was 21, which yielded an accuracy of 72%

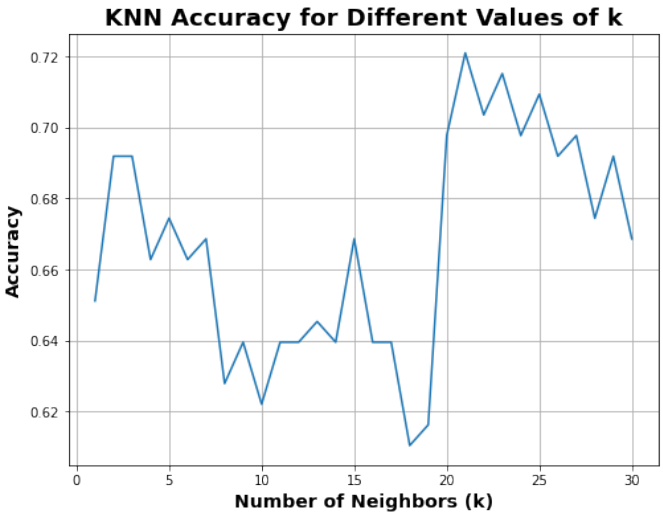

Figure 2: Optimal value of k for KNN algorithm.

Overall this was one of the lower performing models across all success metrics, highlighted by an 78% positive recall, the lowest out of all six models. A common reason for poor performance in a KNN model is from an imbalanced dataset, which was accounted for when the dataset was resampled for the minority case of lung cancer, which leaves another possible reason for this being a dataset with high dimensionality.

C. *Random Forest*

Random forest was a top performing model across all metrics. This was highlighted by a 97% positive recall score, meaning that it only had a 3% false negative rate which again, is of high importance in this use case. Along with logistic regression, the random forest algorithm showed only a minimal drop off in accuracy when compared to positive recall. The cross validation score for random forest was 92.8%, the highest of all models.

D. *Gradient Boosting*

The gradient boosting algorithm had a cross validation score that was in line with many of the other models (89%). An accuracy score of 73% was achieved after performing hyperparameter tuning. This process was used to find the optimal number of estimators to be used in the model to avoid over and under-fitting the model. After searching over several values of estimators, 150 was selected as the number of estimators. Hyperparameter tuning optimized the accuracy of the gradient boosting accuracy by over 10% when comparing the final accuracy of 73% to a 62% accuracy that was shown using an arbitrary value of 100 for the number of estimators.

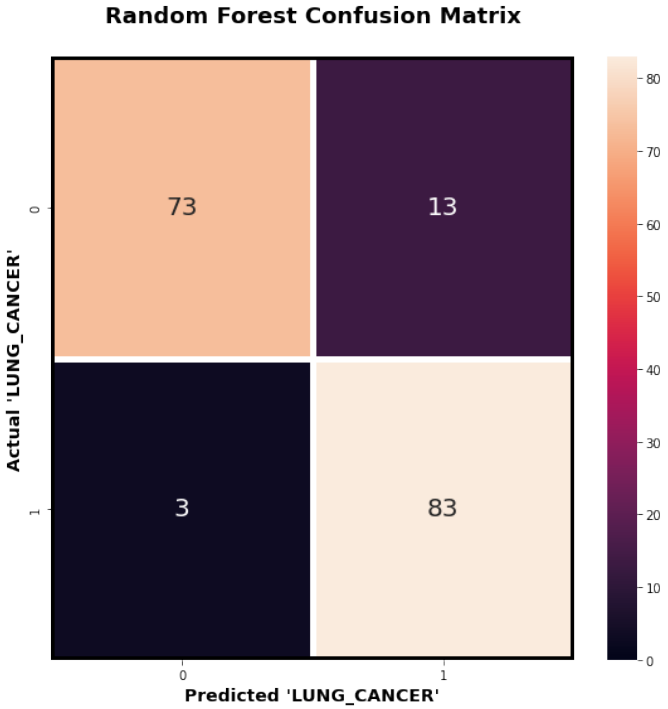

Figure 3: Confusion Matrix for Random Forest algorithm.

E. *Support Vector Machine*

The support vector machine (SVM) model consistently performed much lower than the rest of the of the models even after performing hyperparameter tuning. The accuracy score of 65% was over 10% lower than the next worst model. The highlight of the SVM model was the positive recall score, which was 99%, only generating 1 false negative case. This recall score was increased over 90% after finding the optimal values of C, gamma, and the kernel. Across all models, this metric yielded the largest increase from the score pre-hyperparameter tuning to after the tuning, yet the entirety of the model still has weaknesses when looking at the overall accuracy across all cases.

F. *Decision Tree*

The decision tree model was a high performing model across all metrics. It was one of only two models that had a Positive Case Recall Score, Cross Validation Score, Average Recall Score, and Accuracy of over 80%, showing the versatility of the model and how well it could scale to new test data. The criticism of this model is the max depth that was selected by hyperparameter tuning, which was selected to be 10. This is a relatively steep decision tree given the number of features. While for this test data it doesn't seem to overfit, as the dataset scales to be larger and the test data grows, this would be an area to be aware of to ensure the model contin-

ues to avoid overfitting.

VI. Assessment Methodology

All six models were considered using their most optimal parameters and under the same conditions. Classification reports and confusion matrices were used for all six models to understand strengths, weaknesses and concerns of each. These metrics provide insight into the model's accuracy, sensitivity, and specificity. Cross-validation to validate the model's performance on testing data and to prevent overfitting.

VII. Model Selection

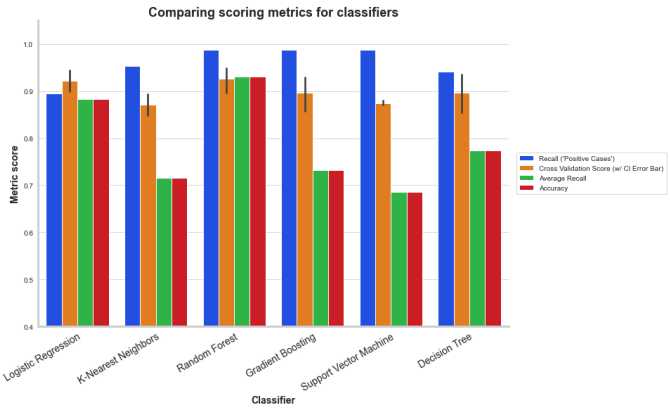

Figure 4: Confusion Matrix for Random Forest algorithm.

After building and optimizing each model, there were two algorithms yield a higher performance than the remaining four: logistic regression and random forest. When looking at the cross validation score, both of these models were at 92%, the highest of any models. While it would be fair to use this metric alone to justify which model will be used as the final selection, however a more thorough investigation can be completed to differentiate the logistic regression and random forest algorithms.

The other metric that will be used to select one of these models is the recall for positive cases. This metric reflects the model's ability to detect true positive cases of lung cancer. The logistic regression algorithm had a positive recall of 88% whereas random forest had a score of 97%. This also means that the random forest model had a lower false negative rate (3% vs 12%), which is the rate at which the the algorithm predicted that a person did not have lung cancer but in reality, they did have lung cancer. False negatives can be a serious issue in medical diagnosis as they can lead to delayed or missed treatment, which can have negative impacts on patient outcomes. A combination of these two, and the remaining success metrics in the graph above, lead to the decision of selecting the random forest algorithm as the final model.

VIII. Ablation

Removing certain components of a machine learning model to measure their individual impact on the overall performance, or an ablation study, is a necessary component to allow for a simple, and often times more effective model. In this case, an ablation study was conducted by removing every combination of the ten features and running an iteration of the final random forest model to measure which combination of features yielded the best result. The best model, with an accuracy score of 97% (6% higher than accuracy using all ten features), removed the variables anxiety, alcohol consuming and coughing. An interesting element of the ablation study was that in the top ten model iterations in terms of accuracy score, nine of them removed the feature 'age' from the data, except the top performing model which left it in the model.

Removed Features	Test Accuracy
ANXIETY', 'ALCOHOL CONSUMING', 'COUGHING'	97.10%
AGE', 'ALCOHOL CONSUMING', 'CHEST PAIN'	95.30%
AGE', 'ALCOHOL CONSUMING'	95.30%
AGE', 'YELLOW_FINGERS', 'ALCOHOL CONSUMING', 'COUGHING'	94.80%
AGE', 'YELLOW_FINGERS', 'ALCOHOL CONSUMING'	94.80%
AGE', 'ANXIETY', 'ALCOHOL CONSUMING', 'COUGHING'	94.80%
AGE', 'ANXIETY', 'ALCOHOL CONSUMING', 'CHEST PAIN'	94.80%
AGE', 'ANXIETY', 'ALCOHOL CONSUMING'	94.80%
AGE', 'YELLOW_FINGERS', 'ANXIETY', 'ALCOHOL CONSUMING', 'COUGHING'	94.20%
AGE', 'WHEEZING', 'ALCOHOL CONSUMING', 'CHEST PAIN'	94.20%

Figure 5: Ablation study removed features.

IX. Tooling

As a final step in this process, a User Interface dashboard was built using Dash by Plotly (Python framework for building web applications that uses Plotly for data visualization), This dashboard will ask a user to answer seven questions that correspond to the seven features that yielded the best random forest accuracy in the ablation study above. Then, using the make_proba function, a probability score is returned that corresponds to the probability that given the user inputs, that user will have been labeled as having lung cancer or not by the random forest model. The output from the dashboard is pictured below.

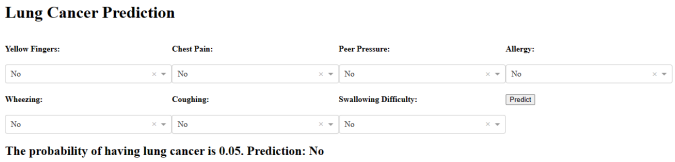

Figure 6: Dashboard using features from ablation.

X. Conclusion and Future Study

As a result of this project, we can successfully classify up to 97% of lung cancer diagnoses using the available features. In practice, the implementation of a dashboard tooling system provides a cheap and precautionary solution to medical practices that can act as prescreening requirements. Future studies of this type of work will be to evolve types of cancers that this algorithm can detect and continue to balance the dataset. Furthermore, the piece of related work "Lung Cancer Classification and Prediction Using Machine Learning and Image Processing" by Nageswaran et al. (2022) includes image processing data. As this lung cancer prediction work scales, there is an opportunity to present a more complex and effective model with the addition of this type of image data, and not rely solely on user responses. The simplicity of this model is adequate for this scale of work, as a more complex model would lead to overfitting, however as this model scales, a more complex model will be required to sufficiently handle differences in response data. The results of this project and all supporting materials came be found on GitHub.

References

[1] Ahmad S. Ahmad, Ali M. Mayya *A new tool to predict lung cancer based on risk factors*, National Library of Medicine (February 26, 2020)

[2] Nageswaran et al. *Lung Cancer Classification and Prediction Using Machine Learning and Image Processing*, National Library of Medicine (August 22, 2022)