CS 4641 Team 116 Final Report

Introduction/Background

Lung cancer is one of the deadliest forms of cancer, with ~75% of patients dying within 5 years of a diagnosis. Early detection is therefore crucial to improve survival rates [1]. Emerging technology using machine-learning (ML) algorithms shows promising results in improving accuracy [2]. Some previously researched algorithms include Logistic Regression with 96.9% and SVM with 99.2% accuracy [2].

Continued research exploring ML models could lead to enhanced accuracy and improved patient outcomes.

This project aims to find the relationship between lung cancer in patients and air pollution to develop a predictive model for lung cancer to increase early identification that can lead to earlier interventions.

1. Dataset Description

This dataset contains information on patients with lung cancer. The features include: demographics, lifestyle, health conditions, risk factors, and symptoms.

2. Dataset Link

Dataset

Problem/Motivation

Cancer is the second leading cause of death in America with lung cancer being the deadliest. Despite this, lung cancer research receives substantially less funding than other cancers. While mortality rates are declining, this is reflective of advances in early detection and advanced treatment [3].

Because early detection is so important in reducing mortality rates, it is imperative that people get screened, especially those at risk. Between a lack of funding and a strong need to combat the deadliest cancer, that is what motivated our proposal. Our model would aim to identify high-risk individuals to encourage them to seek out screenings/treatments for one of the leading causes of death in America.

Machine Learning Methods

1. Data Preprocessing Methods

- **Principal-Component-Analysis (PCA)**: PCA reduces dimensionality and noise in large datasets while keeping optimal variance and underlying patterns. PCA also works to minimize the effects of misleading outliers.
  Library: scikit-learn
  Class: PCA
- **Standardization**: Standardization ensures that each feature would have a mean of 0 and a standard deviation of 1. This ensures that no one feature dominates the cancer-stage prediction.
  Library: scikit-learn
  Class: Standard Scalar
- **Forward Selection**: Forward Selection eliminates unimportant features by training a model to evaluate performance. Features are added individually, and only features with the best performance are kept. This is helpful for datasets with a wide variety of features.
  Library: mlxtend (scikit-learn library extension)
  Class: SequentialFeatureSelector

PCA for Preprocessing

We chose to use Principal Component Analysis (PCA) for several reasons, with the primary goal being to reduce the number of features in our dataset. Initially, our dataset contained 24 features, which, if all were used, could lead to problems such as slow computation, high processing costs, multicollinearity, and overfitting. To address these challenges, we applied PCA to simplify the dataset and improve both computational efficiency and model performance.

We implemented PCA using the python library scikit-learn. First, we removed the columns ‘Index’ and ‘PatientID’ as these columns did not pertain to the feature set. We then standardized the rest of the data (StandardScaler) to ensure that all the features are on the same scale and applied PCA. We then find the covariance matrix, which is useful for figuring out the relationship between variables. After computing the covariance matrix, we found the eigenvalues and eigenvectors associated with each principal component and sorted them in descending order, which allows us to identify the directions with the highest variance. We computed the explained variance and the cumulative sum of the percent of explained variance based on each PCA. Using a 95% variance threshold (>= 0.95), we found that the first 14 PCA’s would reach that threshold. We gathered the top 14 features that contributed most to the explained variance. We dropped the rest of the columns from our dataset and used the modified dataset for our models. Thus, completing the pre-processing for the Naive Bayes model.

2. ML Algorithms/Models

All three specified Algorithms are Supervised.

- **Naive Bayes (NB)**: NB is based on Bayes' Theorem and assumes that features are independent given the class label. It is efficient and performs well on high-dimensional datasets, making it useful for understanding the likelihood of different cancer stages based on feature distributions.
  Library: scikit-learn
  Class: GaussianNB
- **Random Forest (RF)**: RF builds multiple decision trees and combines their output to make a decision. This can help improve accuracy and prevent overfitting. With so many features, RF can help identify key features that contribute to different cancer stages.
  Library: scikit-learn
  Class: RandomForestClassifier

- **Logistic Regression (LR)**: LR focuses on the relationship between each feature and the output (cancer stage). Each coefficient in an LR model would signify the contribution of a specific feature to the likelihood of belonging to a specific cancer stage.
  Library: scikit-learn
  Class: LogisticRegression

Naive Bayes for ML Algorithm

The model we chose to implement on our data was Naive Bayes.

A Naive Bayes model classifier is a probabilistic model that primarily relies on Bayes' Theorem to classify data. The NB model is based on the key assumption that the presence of a particular feature in a class is unrelated to the presence of any other feature.

1. **Model Training and Prediction**
   When the Naive Bayes model was trained with this data, it calculates the probabilities of each feature occurring within each class (cancer stage) based on the training data. Specifically, for each stage, it learns the mean and variance of each feature, assuming a Gaussian distribution. During prediction, the model then calculates the probability of a new patient's feature set belonging to each cancer stage and assigns the patient to the class with the highest probability.

2. **Evaluating the Model**
   This NB model was evaluated using a few key metrics:
   - **Accuracy (0.97):** Accuracy measures the proportion of total predictions the model got correct. It's calculated as the number of correct predictions divided by the total number of predictions. With an accuracy of 0.97 (or 97%), the model is correctly classifying approximately 97% of the cancer stages for new patients. This is a good indication that the model is generally effective at predicting cancer stages.

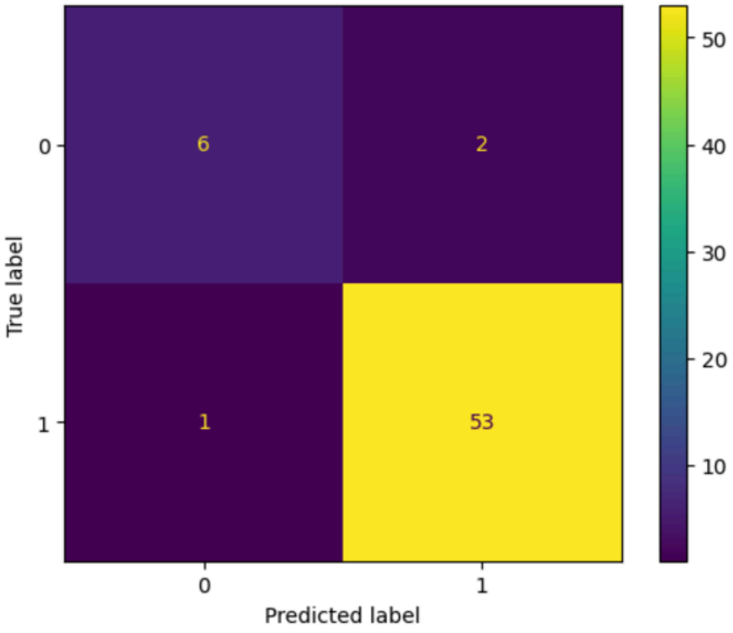
- **Precision Score (0.87 for macro-averaged precision):** Precision measures the accuracy of positive predictions. For each cancer stage, it calculates the proportion of correct positive predictions out of all predictions made for that stage. The macro-averaged precision is the average precision across all classes (Low, Medium, and High). A macro-precision score of 0.87 means that, on average, about 87.00% of the model's positive predictions (predictions for each cancer stage) are correct. High precision indicates that when the model predicts a specific cancer stage, it's often correct, which is valuable for minimizing false alarms in healthcare.
- **F1 Score (0.87 for macro-averaged F1):** The F1 score is the harmonic mean of precision and recall, offering a balance between the two. It’s useful when you want to account for both false positives and false negatives. With an F1 score of 0.87, this metric suggests that the model has a good balance between precision (avoiding false positives) and recall (avoiding false negatives). The model is reasonably consistent in both identifying and correctly classifying cancer stages.
- **Accuracy Score for Each Fold ([0.9355, 0.9355, 0.8710, 0.8548, 0.8525]) in 5-Fold Cross-Validation:**This metric shows the accuracy achieved on each subset of the data during 5-fold cross-validation. Cross-validation is a technique to assess how well the model generalizes unseen data by training and testing it on different subsets of the data. These scores (e.g., 93.55% accuracy on the first fold, 93.55% on the second, etc.) indicate variability in model performance across different portions of the data. The scores are relatively close to one another, which suggests consistent performance across folds, though there is some fluctuation due to differences in data distribution.
- **Average Accuracy Across 5 Folds (0.89)**The average accuracy over the 5 folds gives an overall measure of

the model's performance stability and generalization across different data subsets. With an average cross-validated accuracy of 89%, the model demonstrates strong, consistent performance on different data splits. This consistency provides confidence that the model will likely perform well on new data. The cross-validation scores reveal the model’s performance stability across different subsets of the data, showing how well it generalizes to unseen patients.

- **Recall (0.87 for macro-averaged Recall):** The recall represents how often the Naive Bayes model correctly identifies a true positive in a data set. In this case, we had a 0.87 macro average recall, which means that the model correctly identifies 87% of the actual positive instances across all classes. Looking closer, the recall rate for 0 is 0.75, while the recall rate for 1 is 0.98. This demonstrates that the model is more likely to be able to identify a patient with cancer correctly than without cancer.

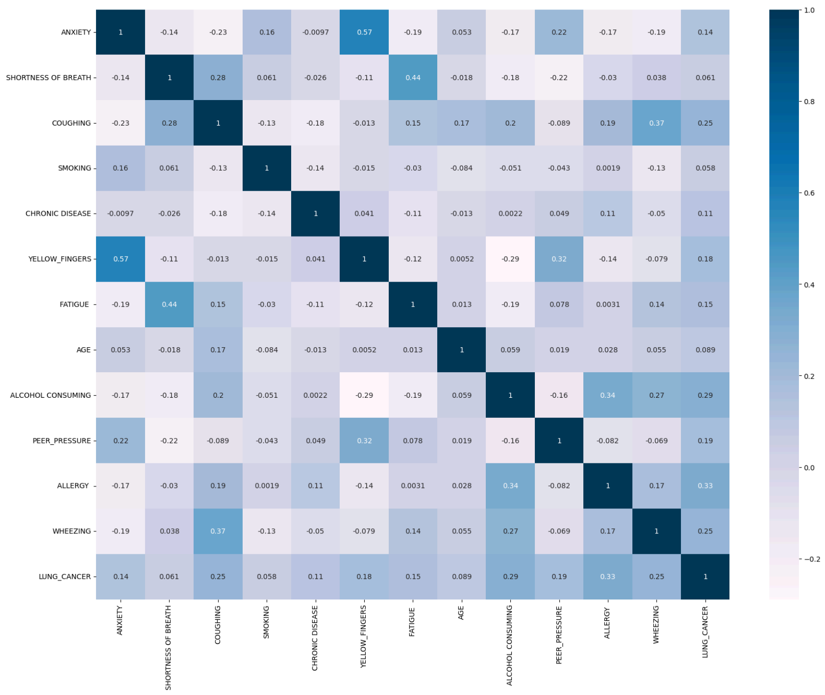
3. **Results and Discussion**

Confusion Matrix



The confusion matrix that was generated shows how often the model misclassified patients across the different stages. By examining the off-diagonal entries, you can see where the model tends to make errors. This helps identify if diagnosing someone with lung cancer or not are harder to predict, which can inform further feature engineering or adjustments to the model.

Correlation Matrix



Our correlation matrix shows low/little correlation between many of our predictor variables, but there are still some variables which are strongly correlated. Naive Bayes operates under the assumption that all predictor variables are conditionally independent from one another. While we do see high correlation between many of the variables, correlation doesn’t necessarily mean causation and there could be other underlying reasons why there is high correlation between those variables. Despite the correlation between some variables, our model performed well with an accuracy of 0.97.

Random Forest for ML Algorithm

Random Forest is an ensemble learning method that combines the predictions of multiple decision trees, each trained on random subsets of data, to improve classification accuracy and reduce overfitting.

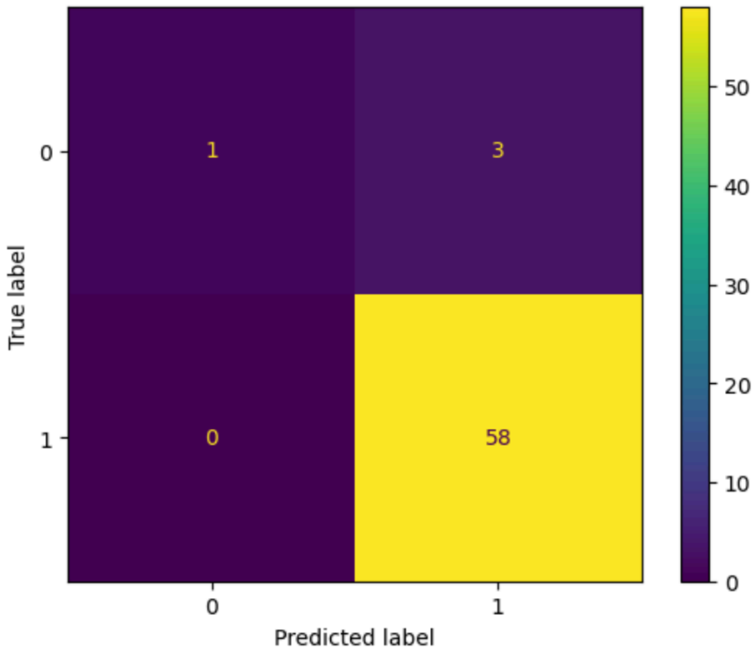
1. **Model Training and Prediction**
   When the Random Forest model was trained with this data, it created an ensemble of decision trees, each built on a random subset of the training data and features. Each tree independently learns a set of decision rules to classify cancer stages, capturing different patterns in the data. During prediction, the model aggregates the outputs from all trees and assigns the new patient's feature set to the cancer stage most frequently predicted by the ensemble.

2. **Evaluating the Model**
   This RF model was evaluated using a few key metrics:
   - **Accuracy (0.90):** Accuracy measures the proportion of total predictions the model got correct. It's calculated as the number of correct predictions divided by the total number of predictions. With an accuracy of 0.90 (or 90%), the model is correctly classifying approximately 90% of the cancer stages for new patients. This is a good indication that the model is generally effective at predicting cancer stages.
   - **Precision Score (0.86 for macro-averaged precision):** Precision measures the accuracy of positive predictions. For each cancer stage, it calculates the proportion of correct positive predictions out of all predictions made for that stage. The macro-averaged precision is the average precision across all classes (Low, Medium, and High). A macro-precision score of 0.86 means that, on average, about 86% of the model's positive predictions (predictions for each cancer stage) are correct. High precision indicates that when the model predicts a specific cancer stage, it's often correct, which is valuable for minimizing false alarms in healthcare.

- **F1 Score (0.78 for macro-averaged F1):** The F1 score is the harmonic mean of precision and recall, offering a balance between the two. It’s useful when you want to account for both false positives and false negatives. With an F1 score of 0.78, this metric suggests that the model has a moderately good balance between precision (avoiding false positives) and recall (avoiding false negatives). The model is reasonably consistent in both identifying and correctly classifying cancer stages.
- **Accuracy Score for Each Fold ([0.9516, 0.9355, 0.9032, 0.8065, 0.9180]) in 5-Fold Cross-Validation:**This metric shows the accuracy achieved on each subset of the data during 5-fold cross-validation. Cross-validation is a technique to assess how well the model generalizes unseen data by training and testing it on different subsets of the data. These scores (e.g., 95.16% accuracy on the first fold, 93.55% on the second, etc.) indicate variability in model performance across different portions of the data. The scores are relatively close to one another with only one drastic drop, which suggests consistent performance across folds. Though there is some fluctuation due to differences in data distribution.
- **Average Accuracy Across 5 Folds (0.90)**The average accuracy over the 5 folds gives an overall measure of the model's performance stability and generalization across different data subsets. With an average cross-validated accuracy of 90%, the model demonstrates strong, consistent performance on different data splits. This consistency provides confidence that the model will likely perform well on new data. The cross-validation scores reveal the model’s performance stability across different subsets of the data, showing how well it generalizes to unseen patients.

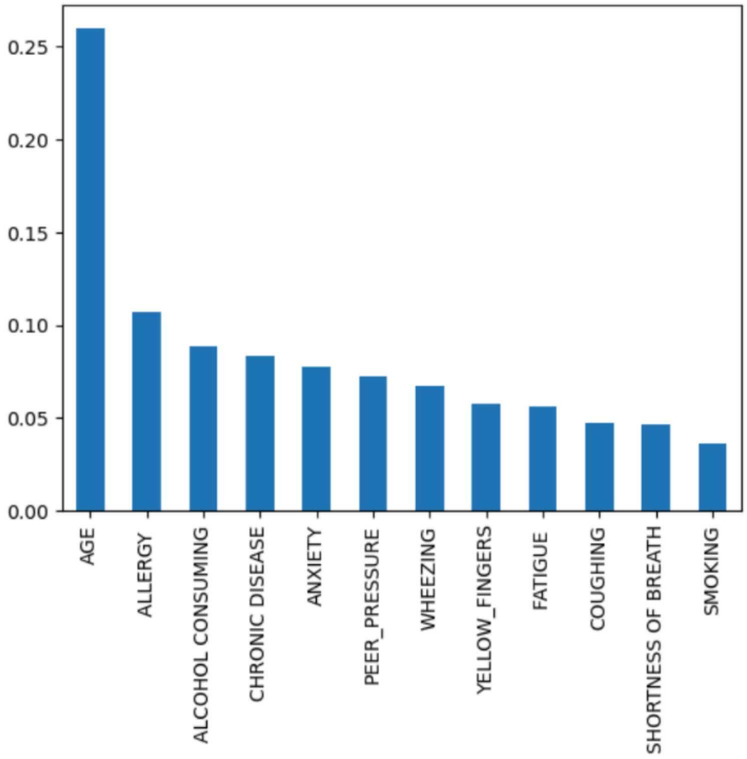
3. **Results and Discussion**

Confusion Matrix



The confusion matrix that was generated shows how often the model misclassified patients across the different stages. By examining the off-diagonal entries, you can see where the model tends to make errors. This helps identify if diagnosing someone with lung cancer or not are harder to predict, which can inform further feature engineering or adjustments to the model.

Feature Importance Bar Chart



This bar chart displays the feature importance scores from a machine learning model, highlighting "Age" as the most influential factor in predicting outcomes, followed by "Allergy" and "Alcohol Consuming."

Logistic Regression for ML Algorithm

Logistic Regression is a statistical model that uses a linear combination of input features passed through a logistic function to estimate the probabilities of class membership for classification tasks.

1. **Model Training and Prediction**
   When the Logistic Regression model was trained with this data, it learned a set of weights for each feature that best separate the cancer stages by fitting a logistic function to the training data. The model calculates the probability of each cancer stage based on the weighted sum of the patient's feature set and assigns the stage with the highest probability as the prediction.

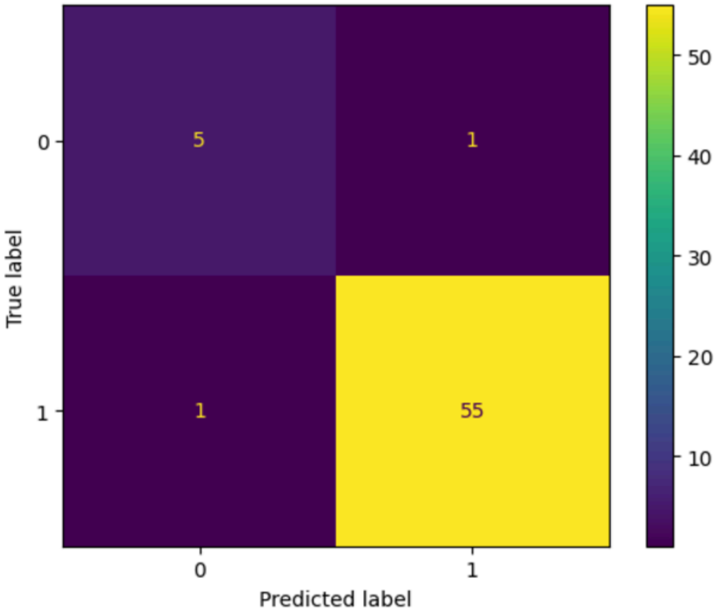
2. **Evaluating the Model**
   This LR model was evaluated using a few key metrics:
   - **Accuracy (0.97):** Accuracy measures the proportion of total predictions the model got correct. It's calculated as the number of correct predictions divided by the total number of predictions. With an accuracy of 0.97 (or 97%), the model is correctly classifying approximately 89.5% of the cancer stages for new patients. This is a good indication that the model is generally effective at predicting cancer stages.
   - **Precision Score (0.91 for macro-averaged precision):** Precision measures the accuracy of positive predictions. For each cancer stage, it calculates the proportion of correct positive predictions out of all predictions made for that stage. The macro-averaged precision is the average precision across all classes (Low, Medium, and High). A macro-precision score of 0.91 means that, on average, about 91% of the model's positive predictions (predictions for each cancer stage) are correct. High precision indicates that when the model predicts a specific cancer stage, it's often correct, which is valuable for minimizing false alarms in healthcare.
   - **F1 Score (0.91 for macro-averaged F1):** The F1 score is the harmonic mean of precision and recall, offering a balance between the two. It’s useful when you want to account for both false positives and false negatives. With an F1 score of 0.91, this metric suggests that the model has a good balance between precision (avoiding false positives) and recall (avoiding false negatives). The model is reasonably consistent in both identifying and correctly classifying cancer stages.
   - **Accuracy Score for Each Fold ([0.9677, 0.9032, 0.8871, 0.8065, 0.918]) in 5-Fold Cross-Validation:**This metric shows the accuracy achieved on each subset of the data during 5-fold cross-validation. Cross-validation is a technique to assess

how well the model generalizes unseen data by training and testing it on different subsets of the data. These scores (e.g., 96.77% accuracy on the first fold, 90.32% on the second, etc.) indicate variability in model performance across different portions of the data. The scores are relatively close to one another, which suggests consistent performance across folds, though there is some fluctuation due to differences in data distribution.

- **Average Accuracy Across 5 Folds (0.90)**The average accuracy over the 5 folds gives an overall measure of the model's performance stability and generalization across different data subsets. With an average cross-validated accuracy of 90%, the model demonstrates strong, consistent performance on different data splits. This consistency provides confidence that the model will likely perform well on new data. The cross-validation scores reveal the model’s performance stability across different subsets of the data, showing how well it generalizes to unseen patients.

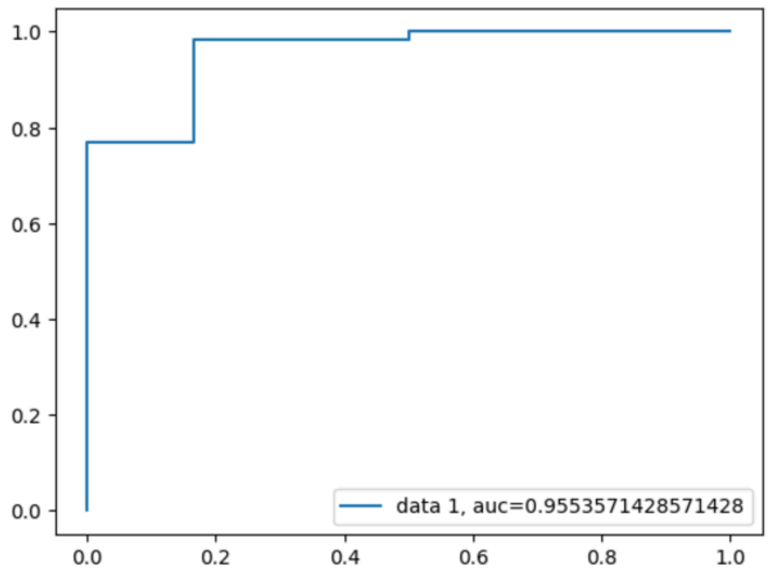
3. **Results and Discussion**

Confusion Matrix



The confusion matrix that was generated shows how often the model misclassified patients across the different stages. By examining the off-diagonal entries, you can see where the model tends to make errors (i.e. predicting Lung Cancer as True when it is actually False). This helps identify if certain stages are harder to predict, which can inform further feature engineering or adjustments to the model.

ROC Curve



This ROC curve illustrates the performance of a logistic regression model, achieving an AUC score of 0.955, indicating a strong ability to distinguish between classes. The closer the curve is to the top-left corner, the better the model's predictive power.

Comparison of Models

The comparison of models highlights the strengths, limitations, and tradeoffs of each approach. The Random Forest model achieved a high recall for the positive class (0.98) and an overall accuracy of 90%. However, it struggled significantly with the negative class, with a precision of 0.80, recall of 0.50, and an F1-score of 0.62. The macro-average metrics, such as recall (0.74) and F1-score (0.78), further indicate imbalanced performance between classes. This poor classification of negatives is likely due to the class imbalance in the dataset, where the minority class (0) had only eight samples. Random Forest, being a majority-voting algorithm, tends to favor the dominant class, and its decision boundaries may have overfit to noise in the minority class data. In contrast, the Naive Bayes model demonstrated balanced precision and recall (both at 0.87), achieving an F1-score of 0.87 and a comparable accuracy of 97%. While its performance was consistent, its lower AUC compared to Logistic Regression suggests reduced overall discriminative ability. This model benefits from its simplicity and effectiveness when features are conditionally independent, although it lacks the flexibility of more advanced models.

Logistic Regression outperformed the other models, achieving the highest precision (0.91), recall (0.91), F1-score (0.91), and AUC (0.955). Its strong, consistent performance across all metrics makes it the most reliable and robust choice. Logistic Regression works particularly well for binary classification when the feature space is relatively linear, which may explain its success here. However, it is less suitable for highly non-linear relationships without additional feature engineering. These results reveal important tradeoffs: Random Forest handles high-dimensional, non-linear data well but struggles with class imbalances; Naive Bayes is simple and effective for small datasets but cannot capture complex relationships between features; and Logistic Regression is a strong baseline that excels in structured, low-dimensional datasets.

Next Steps

To improve Random Forest's performance, addressing the class imbalance is critical. Techniques such as oversampling (e.g., SMOTE), undersampling, or assigning class weights (e.g., class_weight='balanced') can help the model better identify the minority class. Hyperparameter tuning, such as adjusting the number of estimators, maximum depth, and minimum samples per leaf, can also prevent overfitting to noise. Furthermore, combining Random Forest with other models through an ensemble approach, such as boosting, might balance its performance. Collecting additional data for the minority class and revisiting feature engineering to refine the input features would further enhance model performance. Lastly, exploring alternative algorithms like Gradient Boosting models (e.g., XGBoost or LightGBM) could yield improvements, especially in imbalanced datasets. Overall, the results emphasize the importance of evaluating models not only on overall accuracy but also on metrics like precision, recall, and F1-score, especially when class imbalance is present. By addressing these factors, better performance for both classes can be achieved.

References

[1] Yawei Li et al., “Machine Learning for Lung Cancer Diagnosis, Treatment, and Prognosis,” Genomics Proteomics Bioinformatics, vol. 20, no. 5, pp. 850-866, November 2022. [Online]. Available: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC10025752/.

[2] R. K. Pathan et al., "The efficacy of machine learning models in lung cancer risk prediction with explainability," PloS One, vol. 19, no. 6, p. e0305035, Jun. 2024. [Online]. Available: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC11175504/.

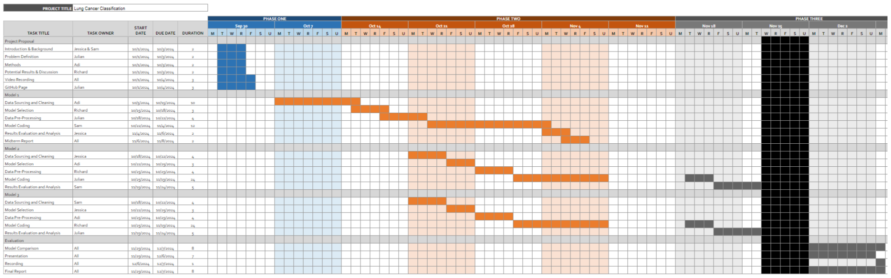
[3] Lung Cancer Research Foundation, “Lung Cancer Facts,” Lung Cancer Research Foundation, 2023. [Online]. Available: https://www.lungcancerresearchfoundation.org/lung-cancer-facts/.

[4] B. Zhang, H. Shi, and H. Wang, “Machine Learning and AI in Cancer Prognosis, Prediction, and Treatment Selection: A Critical Approach,” Machine Learning and AI in Cancer Prognosis, Prediction, and Treatment Selection: A Critical Approach, vol. Volume 16, no. 16, pp. 1779–1791, Jun. 2023. [Online]. Available: https://doi.org/10.2147/jmdh.s410301.

Contribution Table

Name	Contributions
Adi	Data Preprocessing and PCA implementation and model evaluation discussion
Jessica	Naive Bayes Implementation and results discussion
Julian	Random Forest Implementation and results discussion
Richard	Logistic Regression Implementation and results discussion
Sam	Comparison of Models, Next Steps, and Website

Gantt Chart



PROJECT TITLE Lung Cancer Classification

TASK TITLE	TASK OWNER	START DATE	DUE DATE	DURATION
Project Proposal				
Introduction & Background	Jessica & Sam	10/1/2024	10/3/2024	2
Problem Definition	Julian	10/1/2024	10/3/2024	2
Methods	Adi	10/1/2024	10/3/2024	2
Potential Results & Discussion	Richard	10/1/2024	10/3/2024	2
Video Recording	All	10/1/2024	10/4/2024	3
GitHub Page	Julian	10/1/2024	10/4/2024	3
Model 1				
Data Sourcing and Cleaning	Adi	10/5/2024	10/15/2024	10
Model Selection	Richard	10/15/2024	10/18/2024	3
Data Pre-Processing	Julian	10/18/2024	10/22/2024	4
Model Coding	Sam	10/22/2024	11/4/2024	12
Results Evaluation and Analysis	Jessica	11/4/2024	11/6/2024	2
Midterm Report	All	11/6/2024	11/8/2024	2
Model 2				
Data Sourcing and Cleaning	Jessica	10/18/2024	10/22/2024	4
Model Selection	Adi	10/22/2024	10/25/2024	3
Data Pre-Processing	Richard	10/25/2024	10/29/2024	4
Model Coding	Julian	10/25/2024	11/19/2024	24
Results Evaluation and Analysis	Sam	11/19/2024	11/24/2024	5
Model 3				
Data Sourcing and Cleaning	Sam	10/18/2024	10/22/2024	4
Model Selection	Jessica	10/22/2024	10/25/2024	3
Data Pre-Processing	Adi	10/25/2024	10/29/2024	4
Model Coding	Richard	10/25/2024	11/19/2024	24
Results Evaluation and Analysis	Julian	11/19/2024	11/24/2024	5
Evaluation				
Model Comparison	All	11/29/2024	12/7/2024	8
Presentation	All	11/29/2024	12/6/2024	7
Recording	All	12/6/2024	12/7/2024	1
Final Report	All	11/29/2024	12/7/2024	8