Lung Cancer Mortality Prediction

Member 1	Shubham Laxmikant Deshmukh	shubhamd23@vt.edu
Member 2	Spurthi Mohan	spurthim23@vt.edu
Member 3	Ruba Vignesh Balaji	rubavignesh@vt.edu

Introduction

Lung cancer is among the most common types and deadliest forms in many parts of the world. It involves the uncontrolled growth of cells in the lungs, which poses serious health complications and likely death due to ignorance of timely treatment. The prognosis and success in treatment depend on a variety of variables related to the stage at which the diagnosis occurred, combined with the patient's lifestyle choices and any coexisting medical conditions. The demographic and medical backgrounds of these patients, combined with their treatment history, may provide valuable information on the development, course, and response to treatment related to lung cancer. Indeed, the study of trends in patient data has long been the effort of researchers, healthcare professionals, and epidemiologists interested in attempting to improve early detection, optimize treatment plans, and assist with improved survival rates.

In the last couple of decades, various data-driven approaches have been of increased interest for application to healthcare, as a great amount of data on patients is available. Medical researchers can find trends in all three dimensions: patient characteristics, treatment methods applied, and treatment outcomes that may be used for predicting patient survival, finding an optimal treatment plan, and making early interventions for high-risk patients.

Project Problem Statement

Our goal in analysis is to study the factors that affect lung cancer mortality by leveraging a rich dataset of patient information.

The problems are important to a variety of stakeholders because:

- Healthcare Providers: The identification of predictors of survival in lung cancer may allow the physician to individualize treatment and increase the likelihood of cure.
- Medical Researchers: Such correlations between patient factors-smoking status, for example, or BMI, or family history-and cancer prognosis, if identified, may form the basis for new hypotheses in subsequent studies.
- Public Health Organizations: This kind of data regarding survival rates amongst different demographic groups and geographic regions will help immensely in resource allocation, awareness, and prevention drives.

This problem is best framed as an example of a data analytics problem because there is a large amount of available data on patients with multiple influencing variables that create the outcomes. Patterns, correlations, and predictive factors that would have otherwise been laborious and impossible to highlight manually will be made possible with the use of statistical and machine learning techniques. This dataset enables us to study the trend and relationship of numerous variables, namely cancer stage, type of treatment taken, and case history, in relation to patient survival, in order to make informed decisions based on facts and evidence in the healthcare domain.

DataSet

We have finalized the Lung Cancer Mortality Dataset v2 for our project. We found this dataset from kaggle.

The link to this dataset is this: https://www.kaggle.com/datasets/masterdatasan/lung-cancer-mortality-datasets-v2?select=lung_cancer_mortality_data_large_v2.csv

This dataset contains various demographic, medical, and treatment-related variables, allowing for in-depth analysis of factors influencing cancer prognosis and treatment outcomes. Key components of the dataset include:

- **Demographic Information**: Age, gender, and country of residence.
- **Medical History**: Family history of cancer, smoking status, BMI, cholesterol levels, and other health conditions (e.g., hypertension, asthma, cirrhosis).
- **Cancer Diagnosis**: Date of diagnosis and cancer stage.
- **Treatment Details**: Type of treatment received, end treatment date, and survival outcome.
- Target Variable: Did the patient in treatment "Survive" the treatment or not comes under the "Survived" column.

What are the features present & their count, record count?

We read the dataset in a Python notebook, and this is the output of the information of the data frame. The screenshot showcases there are 18 columns/features in the dataset with 9 numerical columns and 9 categorical columns with 3250000 rows/entries.

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 3250000 entries, 0 to 3249999
Data columns (total 18 columns):
 #   Column                      Dtype  
---  ------                      -----  
 0   id                          int64  
 1   age                         float64
 2   gender                      object 
 3   country                     object 
 4   diagnosis_date              object 
 5   cancer_stage                object 
 6   beginning_of_treatment_date  object 
 7   family_history              object 
 8   smoking_status              object 
 9   bmi                         float64
 10  cholesterol_level           int64  
 11  hypertension                int64  
 12  asthma                      int64  
 13  cirrhosis                   int64  
 14  other_cancer                int64  
 15  treatment_type              object 
 16  end_treatment_date          object 
 17  survived                    int64  
dtypes: float64(2), int64(7), object(9)
memory usage: 446.3+ MB
```

Pre-Processing steps:

Pre-processing steps on the Lung Cancer Mortality dataset were very carefully carried out to make the dataset adequate according to the project requirements, clean, balanced, and for analysis and training on various machine learning models. Following are the steps that we performed for pre-processing:

A. Feature Engineering

The dataset was loaded, focusing on the first 700,000 rows to ensure efficient memory usage and manageable data size. An initial exploration of the dataset was performed using df.info(), which provided valuable insights into the dataset's structure, column types, and potential missing values. This step helped identify key areas for improvement during the pre-processing phase.

One of the critical pre-processing steps was handling categorical data. Several columns, such as gender, country, “cancer_stage”, “family_history”, “smoking_status”, and “treatment_type”, were

identified as categorical variables. To make these columns suitable for machine learning models, one-hot encoding was applied. This conversion transformed the categorical data into a numerical format, allowing for seamless integration into the modeling process.

Feature engineering played a pivotal role in enriching the dataset. Using the “Featuretools” library, an “EntitySet” was created to organize the data for advanced feature generation. Aggregation primitives such as mean, sum, and std, along with transformation primitives like day, month, and “is_weekend”, were applied to generate additional features. These newly engineered features provided deeper insights into the data and improved the model's ability to learn patterns. The process was configured to generate up to 250 features with a maximum depth of 2 to balance complexity and usability.

Special attention was given to the target variable, survived. To avoid data leakage, the survived column was temporarily removed during feature engineering and re-added to the final feature matrix. This ensured the integrity of the modeling process while maintaining the relevance of the target variable.

The final feature matrix, containing 225 features, was saved as a CSV file for subsequent analysis and model training. This matrix represented a comprehensive and refined version of the dataset, ready for predictive modeling. Throughout the pre-processing steps, potential issues such as unused primitives due to incompatible columns or insufficient depth were carefully noted, highlighting areas for further optimization.

This systematic approach to pre-processing ensured the dataset was not only clean and balanced but also enriched with meaningful features, providing a strong foundation for the development of accurate and robust machine learning models.

```
<class 'pandas.core.frame.DataFrame'>
Index: 700000 entries, 1 to 700000
Columns: 225 entries, age to survived
dtypes: bool(43), boolean(6), category(9), float64(161), int64(6)
memory usage: 942.2 MB
None
```

- No. Of Entries/ Rows: 700,000
- No. Of features: 225.
- Categorical Columns: 58
- Numerical Columns: 167

B. Exploratory Data Analysis (EDA)

The Exploratory Data Analysis (EDA) process began by checking for missing values in the dataset. Each column was analyzed to calculate the number of NaN values, and the results were sorted in descending order. This helped identify any significant gaps in the data that might require attention. Simultaneously, the number of unique values in each column was assessed to gain insights into data variability and to distinguish between categorical and continuous features.

```
Missing Values per Column:
survived / other_cancer                498133
other_cancer / survived                498133
other_cancer / cirrhosis               495881
cirrhosis / other_cancer               495881
cirrhosis / survived                   422587
                                        ...
cholesterol_level + hypertension            0
cholesterol_level + other_cancer            0
cholesterol_level + survived                0
cirrhosis + hypertension                    0
survived                                    0
Length: 225, dtype: int64
```

Next, the dataset was examined for the presence of positive or negative infinite values. Columns containing such values were identified, as these can cause errors or instability in machine learning models. To address this, the identified columns were dropped from the dataset. This step ensured that the dataset no longer contained invalid data points that could disrupt the analysis or training process.

```
Columns with infinite values removed:
Index(['age / asthma', 'age / cirrhosis', 'age / hypertension',
       'age / other_cancer', 'age / survived', 'asthma / cirrhosis',
       'asthma / hypertension', 'asthma / other_cancer', 'asthma / survived',
       'bmi / asthma', 'bmi / cirrhosis', 'bmi / hypertension',
       'bmi / other_cancer', 'bmi / survived', 'cholesterol_level / asthma',
       'cholesterol_level / cirrhosis', 'cholesterol_level / hypertension',
       'cholesterol_level / other_cancer', 'cholesterol_level / survived',
       'cirrhosis / asthma', 'cirrhosis / hypertension',
       'cirrhosis / other_cancer', 'cirrhosis / survived',
       'hypertension / asthma', 'hypertension / cirrhosis',
       'hypertension / other_cancer', 'hypertension / survived',
       'other_cancer / asthma', 'other_cancer / cirrhosis',
       'other_cancer / hypertension', 'other_cancer / survived',
       'survived / asthma', 'survived / cirrhosis', 'survived / hypertension',
       'survived / other_cancer'],
      dtype='object')
```

After removing columns with infinite values, the dataset was re-evaluated to ensure its integrity. A second check for missing values was performed, and the results confirmed that no null values remained. This reassured us that the dataset was clean and free from incomplete data.

```
Null values in cleaned DataFrame:
age                                  0
bmi                                  0
cholesterol_level                    0
hypertension                         0
asthma                               0
                                    ..
TIME_SINCE(end_treatment_date)       0
YEAR(beginning_of_treatment_date)    0
YEAR(diagnosis_date)                 0
YEAR(end_treatment_date)             0
survived                             0
Length: 190, dtype: int64
```

The next step was to identify and remove features related to the target variable, "survived." Columns containing "survived" in their names (e.g., transformations and interactions like addition, subtraction, or division) were identified. These columns were removed from the feature set (x_cleaned) to prevent data leakage, ensuring that the machine learning model could train effectively without unintended bias.

```
x_cleaned shape: (700000, 163)

y shape: (700000,)

Removed features related to 'survived': ['ABSOLUTE(survived)', 'age + survived', 'asthma +
survived', 'bmi + survived', 'cholesterol_level + survived', 'cirrhosis + survived',
'hypertension + survived', 'other_cancer + survived', 'survived / age', 'survived / bmi',
'survived / cholesterol_level', 'age * survived', 'asthma * survived', 'bmi * survived',
'cholesterol_level * survived', 'cirrhosis * survived', 'hypertension * survived', 'other_cancer
* survived', 'PERCENTILE(survived)', 'age - survived', 'asthma - survived', 'bmi - survived',
'cholesterol_level - survived', 'cirrhosis - survived', 'hypertension - survived', 'other_cancer
- survived', 'survived']
```

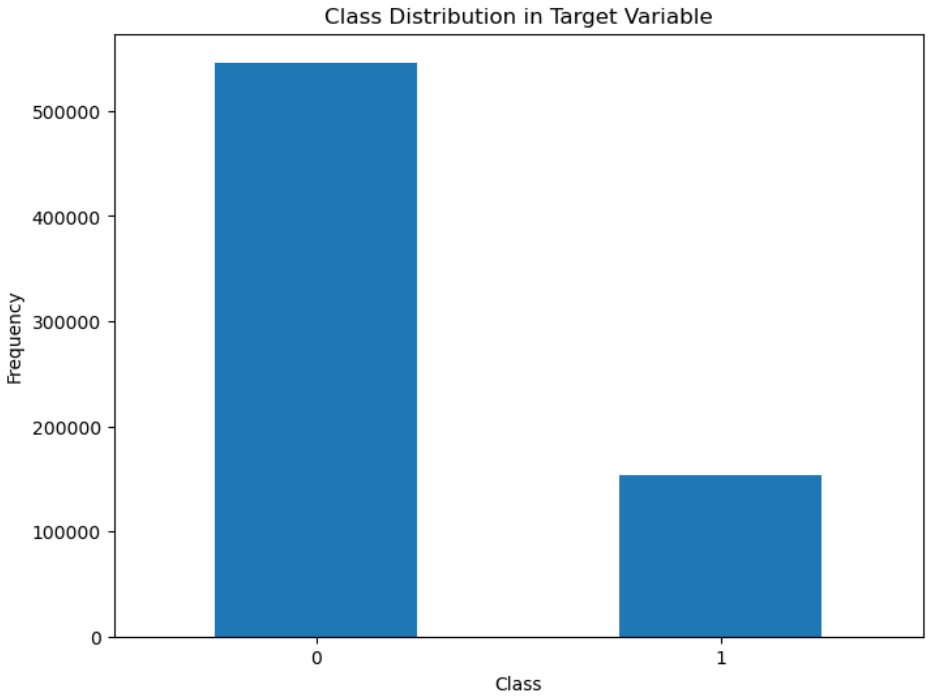
The dataset was then validated to confirm the absence of both missing and infinite values, as well as features related to "survived." At this stage, all columns were consistent and complete, requiring no additional preprocessing. The removal of invalid data points and target-related features ensured the dataset's readiness for further analysis and machine learning tasks.

```
age                                    0
bmi                                    0
cholesterol_level                      0
hypertension                           0
asthma                                 0
                                      ..
TIME_SINCE(diagnosis_date)             0
TIME_SINCE(end_treatment_date)         0
YEAR(beginning_of_treatment_date)      0
YEAR(diagnosis_date)                   0
YEAR(end_treatment_date)               0
Length: 163, dtype: int64
Columns with missing values:
Series([], dtype: int64)
```

Next, the class distribution in the target variable, survived, was analyzed to understand the balance between classes. Both the frequency and percentage distributions were calculated and visualized using a bar chart. This revealed an imbalance in the target variable, which was addressed using Synthetic Minority Oversampling Technique (SMOTE). By applying SMOTE, the class distribution was balanced, ensuring fairness and reliability during model training.

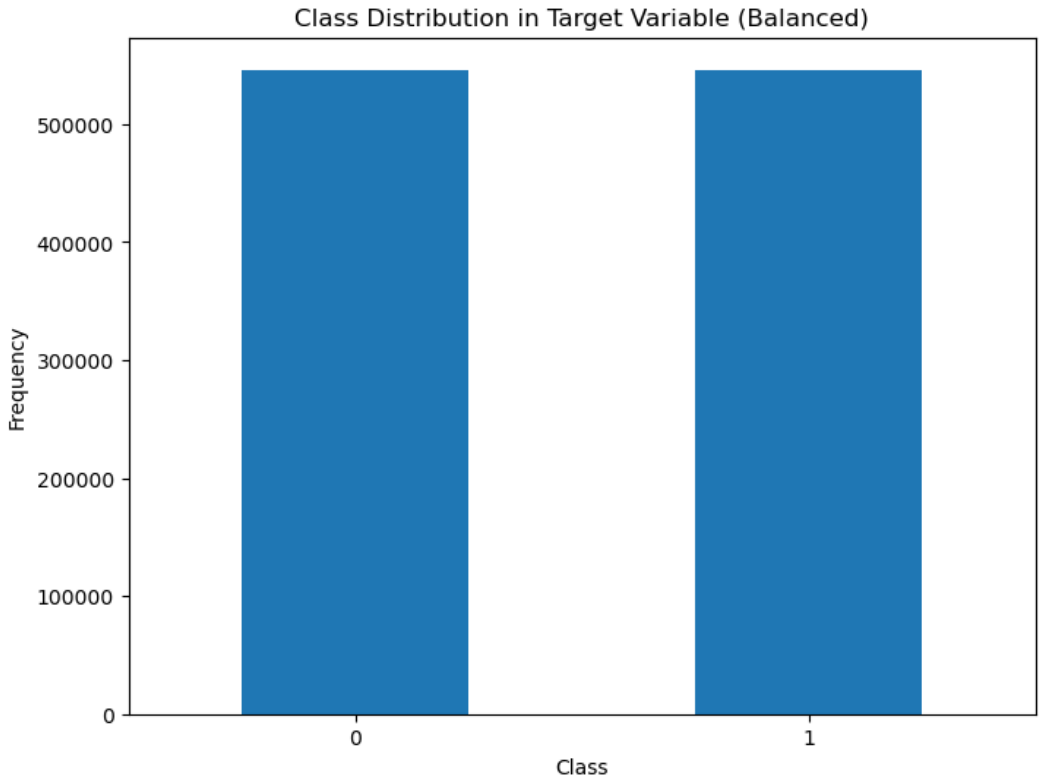
Before Class balancing:

```
Class Distribution in Target Variable:
survived
0    546178
1    153822
Name: count, dtype: int64

Class Percentages:
survived
0    78.025429
1    21.974571
Name: proportion, dtype: float64
```

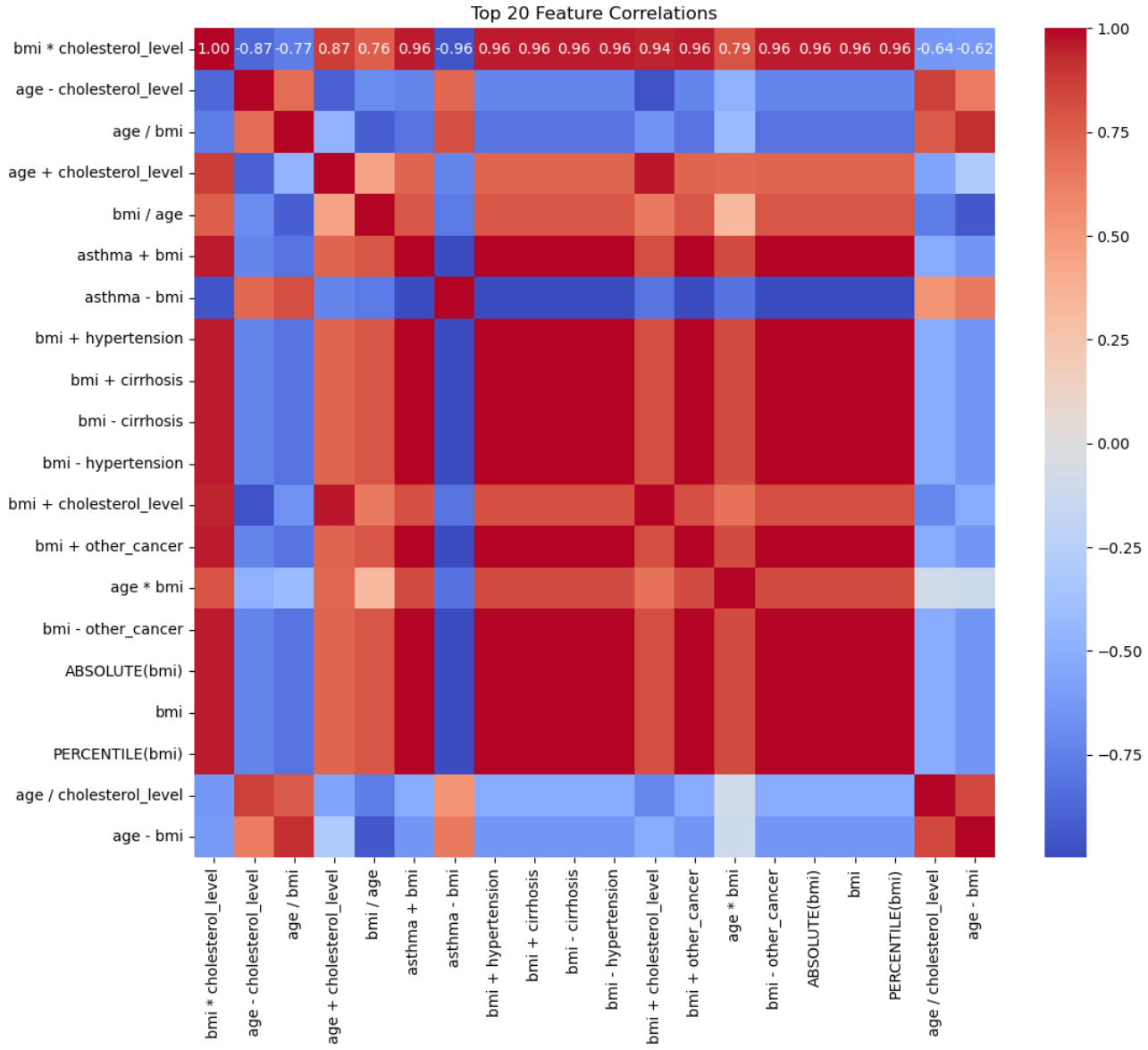


After Class Balancing using SMOTE:

```
Class Distribution After Balancing:
survived
0    546178
1    546178
Name: count, dtype: int64

Class Percentages After Balancing:
survived
0    50.0
1    50.0
Name: proportion, dtype: float64
```



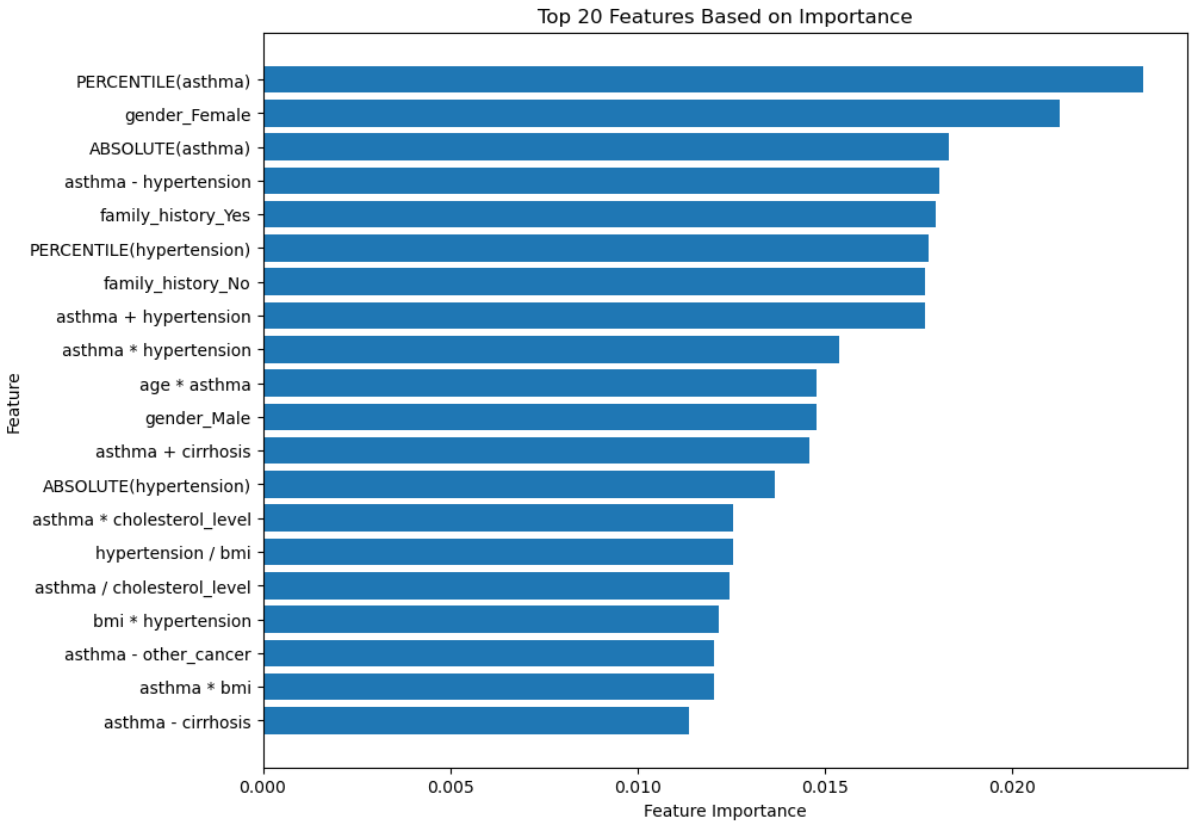
Next, we plotted the correlation heatmap between the columns of the ”x balanced” data frame. The. The heatmap provides a visual representation of the pairwise correlations between the top 20 features in the dataset. It highlights how strongly features are related to one another, with positive correlations shown in warm colors (red) and negative correlations in cool colors (blue). This visualization helps identify patterns, such as highly correlated features that may be redundant or independent features that are likely to provide unique information. By focusing on the top 20 features with the highest overall correlation magnitudes, the heatmap offers a concise view of the most influential relationships in the dataset, aiding in feature selection and dimensionality reduction decisions.



C. Feature Importance

To identify the most important features for modeling, a Random Forest classifier was trained on the balanced dataset. Feature importance scores were extracted, and the top 20 features were identified. These features were visualized using a horizontal bar chart, highlighting their relative importance. This step provided valuable insights into the most predictive features in the dataset.

```
Top 20 Features Based on Importance:
                          Feature  Importance
130          PERCENTILE(asthma)    0.023493
7                 gender_Female    0.021277
51              ABSOLUTE(asthma)    0.018311
145       asthma - hypertension    0.018060
41           family_history_Yes    0.017968
134    PERCENTILE(hypertension)    0.017783
40             family_history_No    0.017696
66        asthma + hypertension    0.017689
117       asthma * hypertension    0.015391
108                age * asthma    0.014800
8                   gender_Male    0.014796
65           asthma + cirrhosis    0.014585
55        ABSOLUTE(hypertension)    0.013663
115  asthma * cholesterol_level    0.012571
94           hypertension / bmi    0.012565
85   asthma / cholesterol_level    0.012471
121          bmi * hypertension    0.012158
146       asthma - other_cancer    0.012055
114                asthma * bmi    0.012031
144          asthma - cirrhosis    0.011376
```

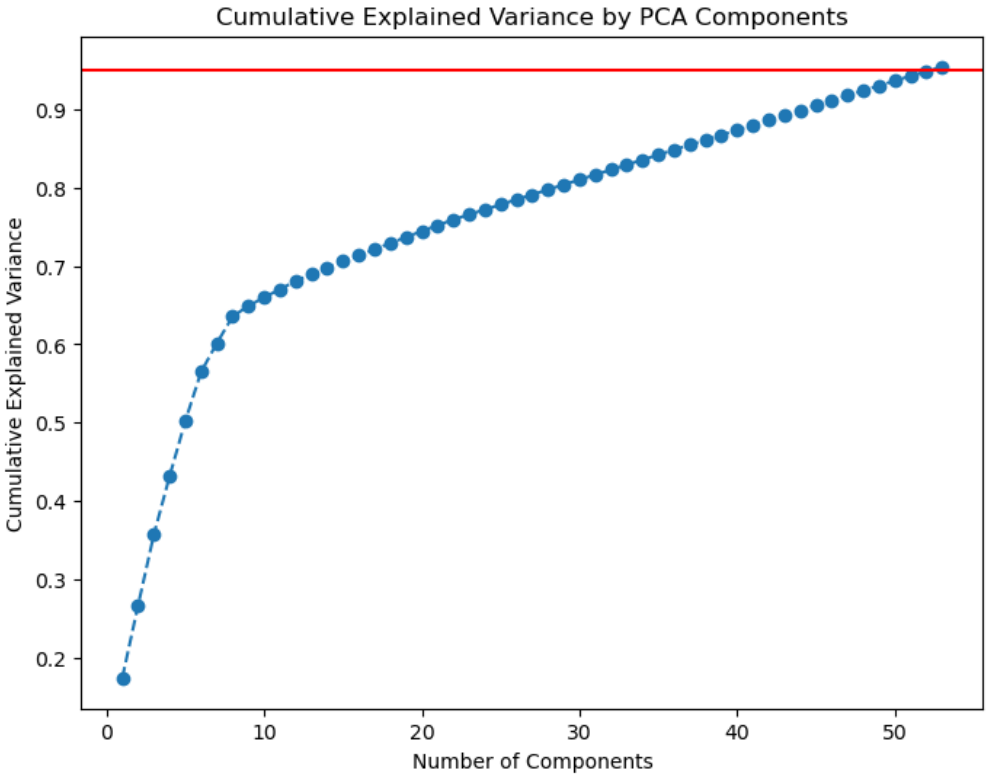


D. Dimensionality reduction using PCA

Principal Component Analysis (PCA) was then applied to reduce the dimensionality of the dataset while retaining 95% of the variance. The features were first standardized using a StandardScaler to ensure consistency. PCA reduced the feature set to a 53 number of components or features, optimizing computational efficiency and enhancing model performance. The number of components selected, and the cumulative explained variance were analyzed and visualized using a line plot.

The EDA process also included a detailed visualization of cumulative explained variance to determine how much variance each principal component contributed. This ensured that the reduced feature set retained sufficient information for accurate predictions while removing redundant or less relevant features.

```
Number of components selected to retain 95% variance: 53
Explained variance ratio for each component:
[0.17423438 0.09242601 0.09023247 0.07460226 0.07083361 0.06247054
 0.03666318 0.03403215 0.01343556 0.01085962 0.01040492 0.01040273
 0.00897127 0.00845391 0.00777743 0.00774248 0.00773106 0.00771507
 0.00769211 0.0076578  0.00740611 0.0073387  0.00639313 0.00638025
 0.00637164 0.00636512 0.00636193 0.00635889 0.00635535 0.00634812
 0.00634554 0.00634397 0.00634084 0.00633772 0.00633631 0.00632837
 0.00632532 0.00632196 0.00632099 0.00631763 0.00631224 0.00630904
 0.00630697 0.00630546 0.00630045 0.00629505 0.00629175 0.00628838
 0.00627011 0.0062078  0.00606578 0.00590247 0.00587791]
Cumulative explained variance:
[0.17423438 0.26666039 0.35689286 0.43149513 0.50232873 0.56479927
 0.60146245 0.6354946  0.64893016 0.65978979 0.67019471 0.68059744
 0.68956871 0.69802262 0.70580005 0.71354253 0.72127359 0.72898865
 0.73668076 0.74433856 0.75174467 0.75908337 0.7654765  0.77185675
 0.77822839 0.78459351 0.79095545 0.79731434 0.80366969 0.81001781
 0.81636334 0.82270732 0.82904815 0.83538588 0.84172219 0.84805056
 0.85437588 0.86069783 0.86701882 0.87333645 0.87964868 0.88595773
 0.89226469 0.89857015 0.9048706  0.91116565 0.9174574  0.92374578
 0.93001588 0.93622369 0.94228946 0.94819193 0.95406984]
```



In conclusion, the EDA process involved cleaning the dataset, balancing the target variable, identifying important features, and reducing dimensionality with PCA. These steps prepared the data for robust and efficient machine learning model development, ensuring high-quality predictions and reliable results.

Methods and Models

A. Methodology

1. Data Splitting

- The dataset was divided into training and testing subsets using **train_test_split** from sklearn.
    - **Training Set**: 80% of the data was used for training the models.
    - **Testing Set**: 20% of the data was reserved for evaluation.
- **Stratification** was used to ensure the class distribution of the target variable (Survived) was maintained across both subsets.

2. Scaling and Normalization

- **Feature Scaling**:
    - For models sensitive to feature magnitudes (e.g., Logistic Regression, KNN, ANN), features were standardized using StandardScaler to ensure they had a mean of 0 and a standard deviation of 1.
- **Normalization**:
    - For distance-based methods like KNN, the dataset was normalized between 0 and 1 to ensure fair distance calculations across features.

3. Model Training

- **Models Used**:
    - **Decision Tree Classifier**
    - **Random Forest Classifier**
    - **Logistic Regression**
    - **Naive Bayes**
    - **XGBoost Classifier**
    - **K-Nearest Neighbors (KNN)**
    - **Artificial Neural Networks (ANN)**

- **Feature Selection Approaches**:
    - **Feature Importance**: Random Forest identified the top features, which were then used to train models.
    - **Dimensionality Reduction (PCA)**: Principal Component Analysis (PCA) was applied to reduce the feature set while retaining 95% of the variance. Models were trained on the reduced feature set.

4. Evaluation Metrics

Performance was assessed using the following metrics:

- **Accuracy**: Measures the percentage of correctly predicted instances.
- **Precision**: Indicates the proportion of true positives among all predicted positives.
- **Recall (Sensitivity)**: Indicates the proportion of true positives among all actual positives.
- **F1 Score**: Harmonic mean of Precision and Recall, providing a balanced view of model performance.
- **AUC (Area Under the Curve)**: Calculated for both ROC and Precision-Recall curves to evaluate overall model performance.

5. Visualization

- **ROC Curve**:
    - False Positive Rate (FPR) vs. True Positive Rate (TPR) was plotted for each model to compare their ability to distinguish between classes.
    - Models such as XGBoost and Random Forest showed higher AUC values, indicating superior classification performance.
- **Precision-Recall Curve**:
    - Precision vs. Recall was plotted to identify the trade-offs for each model, particularly relevant for imbalanced datasets.
    - Models with higher average precision (AP) were preferred for better precision-recall balance.

6. Model Comparison

- Models were evaluated based on their performance with:
    - **Feature Importance**: Using the most important features identified by Random Forest.
    - **PCA**: Using the reduced feature set from PCA.
- Results showed that **Random Forest with PCA** achieved the highest F1 Score (0.8340), followed closely by XGBoost with PCA.

B. Models

The central objective of the project is to classify lung cancer patients into two categories: "Survived" and "Not Survived" after treatment. The target variable is binary, and several machine learning models were applied to tackle this **supervised classification problem**. Each model was chosen based on its unique strengths and ability to handle specific data characteristics. Here's a detailed explanation of each method and model:

1. Decision Tree Classifier

Description: A Decision Tree is a rule-based model that partitions the dataset recursively based on feature values to form a tree-like structure. Each node in the tree represents a decision rule based on a feature, and the leaves represent the predicted class.

How it Works:

- Splits the dataset at each node based on criteria like **Gini Impurity** or **Information Gain**.
- Repeats the process recursively until all leaves are pure (contain only one class) or a stopping criterion is met.

Strengths:

- Highly interpretable and easy to visualize.
- Handles both categorical and numerical features.

Limitations:

- Prone to overfitting, especially with deep trees.
- Sensitive to small changes in data (high variance).

2. Random Forest Classifier

Description: Random Forest is an ensemble learning method that builds multiple decision trees during training. The final prediction is made by averaging the predictions of individual trees (for regression) or by majority voting (for classification).

How it Works:

- Creates multiple bootstrap samples (random subsets) from the dataset.
- Builds a decision tree for each subset.
- Uses random feature selection at each split to reduce correlation among trees.

Strengths:

- Reduces overfitting by averaging multiple models.
- Provides feature importance metrics, allowing insight into which features influence predictions the most.

Limitations:

- Computationally expensive for large datasets.
- Less interpretable than single decision trees.

3. Logistic Regression

Description: Logistic Regression is a statistical method for binary classification that models the relationship between input features and the probability of a binary outcome.

How it Works:

- Computes the weighted sum of input features.
- Applies the **logistic (sigmoid) function** to transform the output into a probability.
- Predicts the class based on a threshold (e.g., 0.5).

Strengths:

- Simple and computationally efficient.
- Interpretable coefficients indicating the impact of each feature.

Limitations:

- Assumes linear separability of data, which may not hold in complex datasets.
- Struggles with multicollinearity among features.

4. Naive Bayes

Description: Naive Bayes is a probabilistic model that applies Bayes' Theorem with the assumption of independence among features.

How it Works:

- Computes the posterior probability of each class given the input features.
- Assumes features are conditionally independent given the class label.
- Assigns the class with the highest posterior probability.

Strengths:

- Fast to train and efficient for high-dimensional data.
- Performs well on small datasets and with categorical data.

Limitations:

- Independence assumption rarely holds in real-world datasets.
- Sensitive to imbalanced class distributions.

5. XGBoost Classifier:

Description: XGBoost (Extreme Gradient Boosting) is a powerful and scalable gradient boosting framework that uses decision trees as base learners and optimizes them using gradient descent.

How it Works:

- Builds an ensemble of weak learners (decision trees) sequentially.
- Minimizes a regularized loss function to improve generalization.
- Uses advanced techniques like shrinkage, column subsampling, and handling missing values for better performance.

Strengths:

- Highly efficient and scalable for large datasets.
- Regularization techniques reduce overfitting.
- Supports parallel processing, making it faster than traditional boosting methods.

Limitations:

- Requires careful tuning of hyperparameters for optimal performance.
- Can be less interpretable than simpler models.

6. K-Nearest Neighbors (KNN)

Description: KNN is a non-parametric model that classifies data points based on their proximity to other data points.

How it Works:

- Computes the distance between the test point and all training points.
- Identifies the k nearest neighbors.
- Assigns the class label based on the majority class among the neighbors.

Strengths:

- Simple and intuitive.
- Does not make assumptions about data distribution.

Limitations:

- Computationally expensive for large datasets due to distance calculations.
- Sensitive to irrelevant features and scaling.

7. Artificial Neural Networks (ANN)

Description: ANN is a deep learning model inspired by the human brain. It consists of layers of interconnected neurons that transform inputs into outputs through non-linear functions.

How it Works:

- Input features are passed through multiple layers.
- Each neuron applies a weighted sum followed by an activation function (e.g., ReLU, sigmoid).
- The output layer provides predictions.

Strengths:

- Capable of modeling complex, non-linear relationships.
- Highly flexible and scalable for large datasets.

Limitations:

- Computationally intensive.
- Requires careful tuning of hyperparameters (e.g., learning rate, hidden layers).

Comparison of Methods

Each model addresses specific challenges:

- **Decision Tree**: Offers interpretability but risks overfitting.
- **Random Forest**: Balances bias and variance, ideal for feature importance analysis.
- **Logistic Regression**: Provides a simple, interpretable baseline.
- **Naive Bayes**: Effective for smaller datasets and simple distributions.
- **XGBoost:** Combines efficiency, scalability, and regularization for high-performing models on large datasets.
- **KNN**: Relies on spatial relationships, sensitive to scaling.
- **ANN**: Excels at capturing non-linear patterns but requires computational resources.

Results

We compared two feature selection methods, **Feature Importance** and **PCA (Principal Component Analysis)**, by applying them to various machine learning models, including **Decision Tree**, **Random Forest**, **Logistic Regression**, **Naive Bayes**, **XGBoost**, **KNN**, and **ANN**. The models were evaluated using key performance metrics such as **Accuracy**, **Precision**, **Recall**, **F1 Score**, and **AUC (Area Under the Curve)**. Additionally, visualizations, including **ROC Curves** and **Precision-Recall Curves**, were utilized to provide deeper insights into the models' performance and their ability to handle class imbalance effectively.

A. Feature Importance

These are the results of classification reports of the 7 Machine Learning models using feature importance:

- **Decision Tree:**

Decision Tree Performance:

	precision	recall	f1-score	support
0	0.77	0.89	0.82	163615
1	0.87	0.74	0.80	164092
accuracy			0.81	327707
macro avg	0.82	0.81	0.81	327707
weighted avg	0.82	0.81	0.81	327707

- **Random Forest:**

Random Forest Performance:

	precision	recall	f1-score	support
0	0.77	0.97	0.86	163615
1	0.95	0.71	0.82	164092
accuracy			0.84	327707
macro avg	0.86	0.84	0.84	327707
weighted avg	0.86	0.84	0.84	327707

- **Logistic Regression:**

Logistic Regression Performance:

	precision	recall	f1-score	support
0	0.68	1.00	0.81	163615
1	1.00	0.54	0.70	164092
accuracy			0.77	327707
macro avg	0.84	0.77	0.76	327707
weighted avg	0.84	0.77	0.76	327707

- **Naive Bayes:**

Naive Bayes Performance:

	precision	recall	f1-score	support
0	0.70	0.77	0.73	163615
1	0.74	0.67	0.70	164092
accuracy			0.72	327707
macro avg	0.72	0.72	0.72	327707
weighted avg	0.72	0.72	0.72	327707

- **XGBoost Classifier:**

XGBoost Performance:

	precision	recall	f1-score	support
0	0.77	1.00	0.87	163615
1	1.00	0.70	0.82	164092
accuracy			0.85	327707
macro avg	0.88	0.85	0.85	327707
weighted avg	0.88	0.85	0.85	327707

- **KNN classifier:**

K-Nearest Neighbors (KNN) Performance:

	precision	recall	f1-score	support
0	0.76	0.93	0.84	163615
1	0.91	0.71	0.80	164092
accuracy			0.82	327707
macro avg	0.84	0.82	0.82	327707
weighted avg	0.84	0.82	0.82	327707

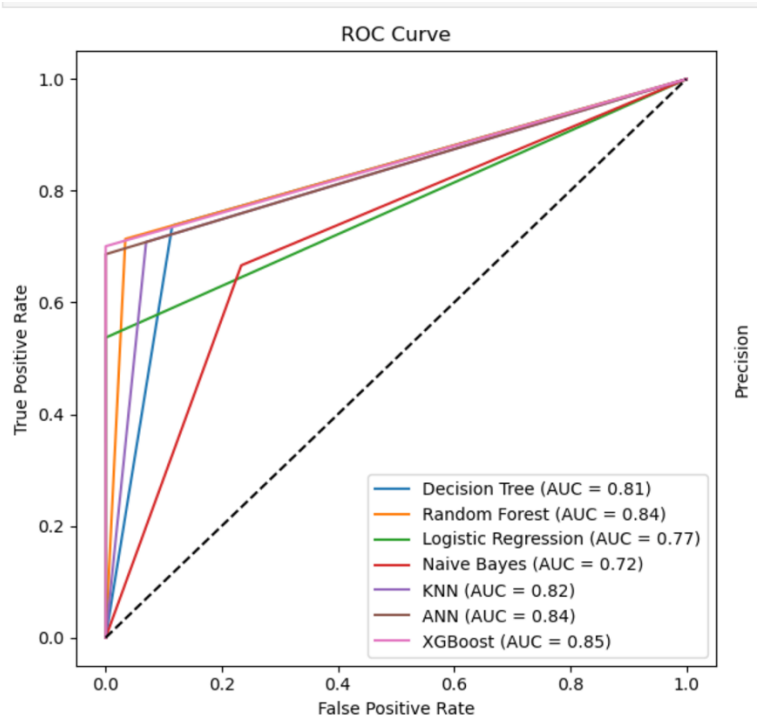
- **Artificial Neural Network:**

ANN Performance:

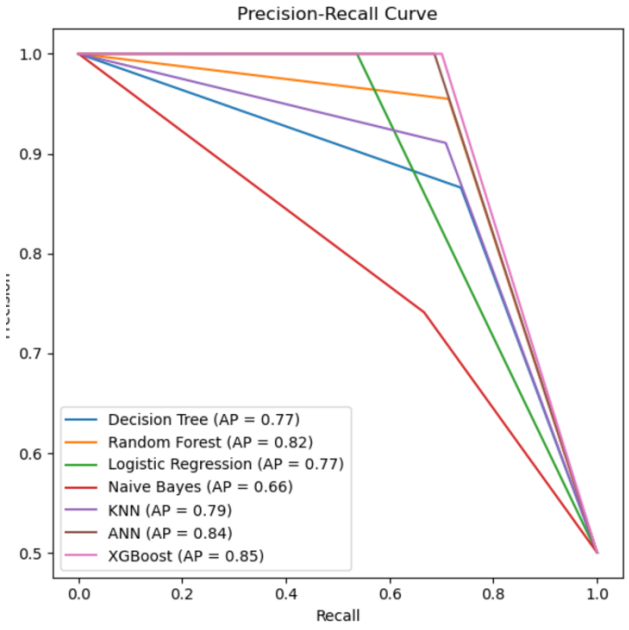
	precision	recall	f1-score	support
0	0.76	1.00	0.86	163615
1	1.00	0.69	0.81	164092
accuracy			0.84	327707
macro avg	0.88	0.84	0.84	327707
weighted avg	0.88	0.84	0.84	327707

The graphs with the performance of all the ML models:

- **ROC Curve:**



- **Precision- Recall Curve:**



B. PCA

These are the results of classification reports of the 7 Machine Learning models using PCA:

- **Decision Tree:**

Decision Tree Performance:

	precision	recall	f1-score	support
0	0.78	0.76	0.77	163615
1	0.76	0.78	0.77	164092
accuracy			0.77	327707
macro avg	0.77	0.77	0.77	327707
weighted avg	0.77	0.77	0.77	327707

- **Random Forest:**

Random Forest Performance:

	precision	recall	f1-score	support
0	0.78	1.00	0.87	163615
1	1.00	0.72	0.83	164092
accuracy			0.86	327707
macro avg	0.89	0.86	0.85	327707
weighted avg	0.89	0.86	0.85	327707

- **Logistic Regression:**

Logistic Regression Performance:

	precision	recall	f1-score	support
0	0.75	0.80	0.78	163615
1	0.79	0.74	0.76	164092
accuracy			0.77	327707
macro avg	0.77	0.77	0.77	327707
weighted avg	0.77	0.77	0.77	327707

- **Naive Bayes:**

Naive Bayes Performance:

	precision	recall	f1-score	support
0	0.77	0.97	0.86	163615
1	0.96	0.72	0.82	164092
accuracy			0.85	327707
macro avg	0.87	0.85	0.84	327707
weighted avg	0.87	0.85	0.84	327707

- **XGBoost Classifier:**

XGBoost Performance:

	precision	recall	f1-score	support
0	0.78	1.00	0.87	163615
1	1.00	0.72	0.83	164092
accuracy			0.86	327707
macro avg	0.89	0.86	0.85	327707
weighted avg	0.89	0.86	0.85	327707

- **KNN classifier:**

K-Nearest Neighbors (KNN) Performance:

	precision	recall	f1-score	support
0	0.73	0.92	0.82	163615
1	0.90	0.66	0.76	164092
accuracy			0.79	327707
macro avg	0.81	0.79	0.79	327707
weighted avg	0.81	0.79	0.79	327707

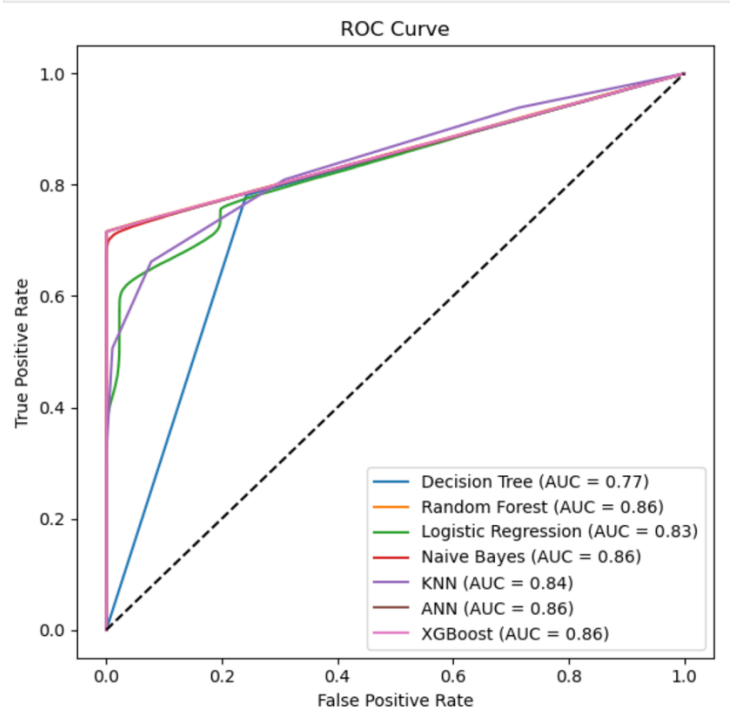
- **Artificial Neural Network:**

ANN Performance:

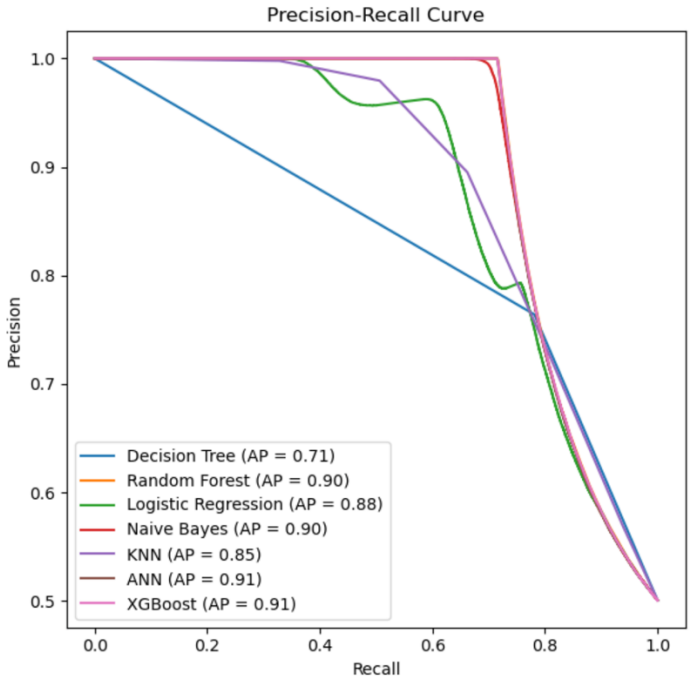
	precision	recall	f1-score	support
0	0.78	1.00	0.87	163615
1	1.00	0.72	0.83	164092
accuracy			0.86	327707
macro avg	0.89	0.86	0.85	327707
weighted avg	0.89	0.86	0.85	327707

The graphs with the performance of all the ML models:

- **ROC Curve:**



- **Precision- Recall Curve:**



C. Comparision of all Models with respect to Feature Importance and PCA methods:

Model Comparison Table: Feature Importance vs PCA

Model	Method	Accuracy	Precision	Recall	F1 Score
Decision Tree	Feature Importance	0.8114	0.8659	0.7375	0.7966
Random Forest	Feature Importance	0.8398	0.9548	0.7138	0.8169
Logistic Regression	Feature Importance	0.7682	1.0000	0.5371	0.6988
Naive Bayes	Feature Importance	0.7164	0.7411	0.6665	0.7018
KNN	Feature Importance	0.8191	0.9107	0.7083	0.7968
ANN	Feature Importance	0.8430	1.0000	0.6864	0.8140
XGBoost	Feature Importance	0.8501	0.9999	0.7007	0.8240
Decision Tree	PCA	0.7696	0.7640	0.7813	0.7725
Random Forest	PCA	0.8571	0.9973	0.7167	0.8340
Logistic Regression	PCA	0.7690	0.7890	0.7354	0.7613
Naive Bayes	PCA	0.8450	0.9632	0.7179	0.8227
KNN	PCA	0.7919	0.8951	0.6620	0.7611
ANN	PCA	0.8569	0.9971	0.7162	0.8336
XGBoost	PCA	0.8570	0.9983	0.7157	0.8337

According to the best F1 Score the best ML model out of the 14 ML models is:

```
Best Combination:
Model: Random Forest
Method: PCA
F1 Score: 0.8340
```

Conclusion

The **Lung Cancer Mortality Prediction project** represents a significant step forward in applying machine learning to healthcare challenges, particularly for a critical issue like lung cancer, which is among the leading causes of mortality worldwide. This project successfully harnessed a diverse and rich dataset containing demographic, medical, and treatment-related information to predict patient survival and provide actionable insights into the factors that influence outcomes.

Key Contributions and Insights

1. **Data Utilization and Preparation**: The project leveraged a comprehensive dataset containing 225 features, encompassing variables such as patient demographics, smoking status, BMI, cancer stage, treatment details, and survival outcomes. The preprocessing phase was meticulous, including cleaning, handling missing values, balancing the target variable using SMOTE, and transforming categorical data into machine-readable formats via one-hot encoding. The inclusion of advanced feature engineering techniques added

depth to the dataset by creating new features, enriching the data, and providing a strong foundation for predictive modeling.

2. **Dimensionality Reduction and Feature Importance**: To address the high dimensionality of the dataset, the project applied Principal Component Analysis (PCA), which reduced the feature set to 53 components while retaining 95% of the variance. This step was crucial in improving computational efficiency and model performance. Additionally, the feature importance analysis performed using Random Forest revealed the most predictive features, offering valuable insights into the relationships between patient characteristics and survival outcomes. This identification of critical features aids in focusing medical attention on the most influential factors.

3. **Model Performance and the Choice of F1 Score**: Multiple machine learning models were tested, including Decision Trees, Random Forest, XGBoost, Logistic Regression, Naive Bayes, K-Nearest Neighbors (KNN), and Artificial Neural Networks (ANN). Among these, **Random Forest combined with PCA** emerged as the best-performing model with an F1 score of 0.8340, followed closely by **XGBoost with PCA**.

The **F1 score** was chosen as the primary metric to evaluate model performance because it provides a balanced measure of **precision** (the ability of the model to avoid false positives) and **recall** (the ability to correctly identify true positives). This was especially important in this project due to the **imbalanced dataset**, where the "Survived" class was underrepresented. Traditional metrics like accuracy would not adequately capture the model's performance on the minority class, as a high accuracy score could still be achieved by simply predicting the majority class. The F1 score, as the harmonic mean of precision and recall, ensures that both false positives and false negatives are penalized, making it a more reliable measure for this type of classification problem. Visualization techniques, such as ROC and Precision-Recall curves, further validated the robustness of the models.

4. **Actionable Insights**: The project's findings have far-reaching implications. The identification of critical features, such as cancer stage, treatment type, and smoking status, provides actionable insights that healthcare providers can use to individualize treatment plans and prioritize high-risk patients. Public health organizations can also leverage these insights to allocate resources effectively, target awareness campaigns, and design prevention programs for populations at risk. Furthermore, researchers can use the identified predictors to formulate new hypotheses and advance the study of lung cancer prognosis and treatment.

Healthcare and Technological Implications

This project showcases the transformative potential of machine learning in healthcare. By integrating predictive models into clinical workflows, healthcare providers can enhance early

detection, optimize treatment strategies, and ultimately improve patient outcomes. The ability to predict survival outcomes with high accuracy also supports decision-making in resource allocation, ensuring that medical resources are directed where they are most needed.

Moreover, the project's use of advanced techniques, such as PCA for dimensionality reduction and Random Forest for feature importance analysis, highlights the importance of combining statistical and machine learning methods to extract meaningful patterns from large and complex datasets. These techniques can be extended to other diseases and healthcare challenges, paving the way for a broader impact of data-driven healthcare solutions.

Future Scope

The project lays the foundation for further research and development in predictive healthcare analytics. Future work could involve:

- Exploring ensemble methods to combine the strengths of different models for even better performance.
- Integrating real-time data to enhance the relevance and timeliness of predictions.
- Investigating causal relationships between features to uncover deeper insights into lung cancer prognosis.
- Extending the approach to other forms of cancer or chronic diseases, creating a generalizable framework for mortality prediction.

Conclusion

The Lung Cancer Mortality Prediction project is a testament to the growing synergy between data science and healthcare. By addressing a critical healthcare challenge with innovative machine learning methods, the project has demonstrated the ability to provide actionable insights that can save lives and improve the quality of care. The selection of the F1 score as the key evaluation metric reflects the project’s commitment to addressing the challenges posed by imbalanced data and ensuring equitable model performance across all classes. This work not only contributes to the field of cancer research but also establishes a robust framework for tackling other complex health issues through data-driven approaches. With its ability to empower healthcare providers, researchers, and policymakers, this project underscores the immense potential of machine learning to transform the healthcare landscape.

Work Distribution

Member 1: Shubham Laxmikant Deshmukh	Preprocessing the data according to the project instructions + EDA + Model Training on PCA reduced data.
Member 2: Spurthi Mohan	Preprocessing the data according to the project instructions + EDA + Model Training on Feature importance reduced data.
Member 3: Ruba Vignesh Balaji	Preprocessing the data according to the project instructions + EDA + Model Training.