COLLATION OF CANCER IN LUNGS UTILIZING MLOPS STRATEGIES

Abstract

After witnessing COVID-19, the gravity of the issue of early detection of a disease is apparent. Lung cancer has proven to be a prominent factor for death worldwide, with an alarming rate of about five million fatal cases per year. In lung cancer, the cancerous cells in the lungs multiply uncontrollably. Three popularly used classifiers for data analytics viz. Logistic Regression, Decision Tree and Neural Networks, were used to analyze lung cancer detection. This work revolves around the objective of investigating the effectiveness of classification algorithms in the early identification of lung cancer. This work has made use of a lung cancer dataset from an online cancer detection system in the public domain. After efficient data mining, classification models have been trained for lung cancer prediction. The experiment is performed on Azure Machine Learning Studio, Microsoft's development environment.

Keywords: Classification, decision tree, logistic regression, lung cancer, machine learning, neural network

Authors

Hitesh Kumar
Department of Artificial Intelligence
Amity School of Engineering and Technology
Amity University, Noida, Uttar Pradesh, India.
https://orcid.org/0000-0001-9061-4388
https://scholar.google.com/citations?user=OE1IYIoAAAAJ
hiteshb3004@gmail.com

Nikita Narwat
Department of Artificial Intelligence
Amity School of Engineering and Technology
Amity University, Noida, Uttar Pradesh, India.
nikitanarwat66@gmail.com

Madhulika Bhatia
Amity School of Engineering and Technology
Amity University, Noida, Uttar Pradesh, India.
madhulikabhatia@gmail.com

I. INTRODUCTION

Lung cancer has been major contributor for the mortality rate around the globe. Windpipe, major airway, or lungs are the initiation sites for the attack by lung cancer. Unregulated proliferation of malignant cells from the lungs. People diseased with emphysema and chronic bronchitis, or a history of chest difficulties are often accompanied with lung cancer. Smoking is a prevailing factor for lung cancer in Indian men; however, smoking is less observed in women of India, implicating that there are other causes that contribute to the spread of lung cancer. The workplace subjection to Radon gas, air pollution, and toxins, are all risk factors. The incubation period of lung cancer is long and are often diagnosed within the age bracket 55 to 65 [1]. As per the studies, work habits and social habits are also a major factor in the aggravation of the disease. A greater awareness of risk factors can aid in the prevention of lung cancer. Early discovery of the disease eases the process of treatment thereby increasing the survival rate of the patient. Early discovery plays a vital role in decreasing mortality rate by utilizing machine learning (ML) approaches, and if this can be used to make the diagnosis more methodical for radiologists, it will lead to early detection.
Lungs are the major site for primary lung cancer whereas in secondary lung cancer, cancerous cells developed in lung are propagated to other body organs. The measure and expansion of the tumor in body establishes its stage.

Kaggle Repository is the source of the dataset employed in this analysis. After preprocessing of the data, and the dataset is split into train and test data. The classification models such as logistic regression, boosted decision trees, and neural networks are applied over the lung cancer dataset having 2 class as the target variable values. To determine the weights of the model, they are trained over the training data. Then, they are scored on different performance measures like F1 score, recall for accuracy evaluation, by testing them over testing data. Finally, using the accuracy metrics, the different classifiers can be compared for the optimal selection of the classification model for the given dataset. For overview of the experiment, refer to fig 1.

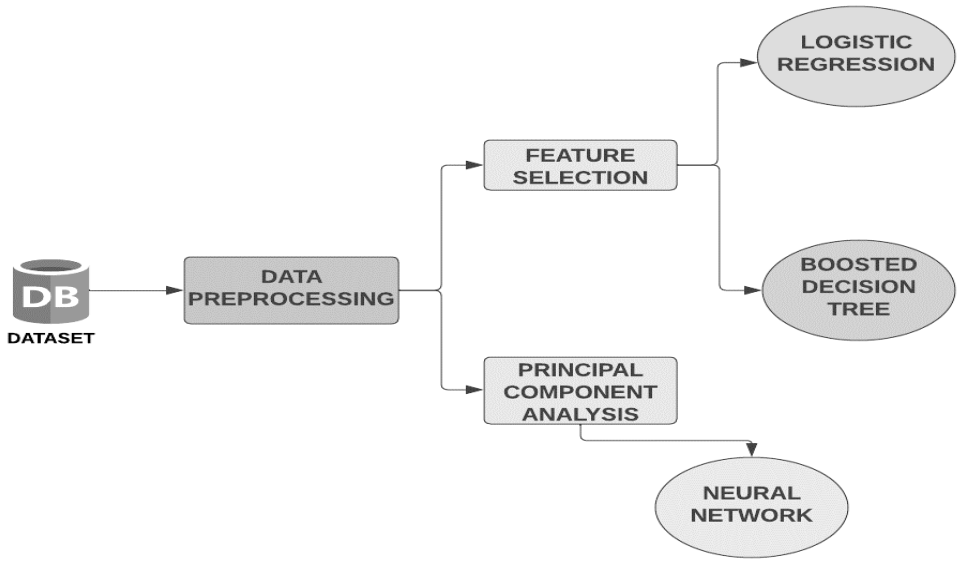


Figure 1: Overview of Experiment

II. RELATED WORK

Many researchers have contributed to numerous lung cancer prediction and classification studies. Using predictive data mining methods, Danjuma [2] evaluatesalgorithms like Decision Trees, Naive Bayes, and Artificial Neural Networks to determine how longlung cancer patients can expect to live following surgery. The algorithms were subjected to a stratified 10-fold cross-validation analysis, and the models' accuracy metrics were assessed.

With the collected lung cancer dataset, Zehra et al. [3] produced various outcomes for each classifier. Following the implementation of the classifiers, comparable accuracy rates for KNN, SVM, NN, and Logistic Regression were discovered. With 99.3% accuracy, Support Vector Machine comes in first. Doctors were able to make more accurate judgements thanks to the proposed method's application to the medical dataset. Ada et al. [4] Naive Bayes, the Hidden Markov Model, and other segmentation techniques were discussed. Several segmentation algorithms used to find lung cancers are properly explained in terms of how and why they work.

Yu et al. [5] used the C4.5 classifier[6] and several feature selection procedures to categorise the type of lung cancer in the Weka environment [7]. Badjio et al. [8] used different feature selection techniques with K-nearest neighbour classifiers and it was implemented as IBK [9] in the Weka environment. Avci et al. [10] suggested a general discriminant analysis (GDA) and most tiny square support vector machine (LS-SVM) based classifier to handle this low sample, high-dimension classification challenge. For classifying lung cancer, Tan et al. [11] integrated the idea of the smallest message length with an indirect decision tree inference process.

With the emergence of deep learning, it has been discovered that autoencoders and other techniques can be used to determine the underlying structure of data. Syed et al. [12] provide a deep autoencoder classification process that first learns deep features before inputting these features into training an artificial neural network. According to experimental findings, when taught with identical training samples and all attributes, the deep learning classifier beats all other classifiers. Performance enhancement is also shown to be statistically significant.

III. MODEL DEPLOYMENT ENVIRONMENT

1. **Azure Machine Learning Studio:** The technology used for the comparative analysis is Azure ML Studio. Thanks to Azure Machine Learning, data scientists and developers can design, deploy, and manage high-quality models quickly and confidently. With industry-leading machine learning operations (MLOps), open-source interoperability, and integrated tools, it reduces time to value. This reputable platform was created for ethical AI machine-learning applications [13]. On Azure ML Studio, the models were trained for the same dataset using Two-Class Boosted Decision Tree, Two-Class Logistic Regression, and Two-Class Neural Network. The results were then compared.

IV. METHODOLOGY

1. **Dataset Description:** This study made use of a Lung Cancer dataset named "Survey Lung Cancer Dataset" from the Kaggle Repository [14]. This dataset has already been utilized in several lung cancer prediction and analysis algorithms. The dataset has 16 total number of attributes and 284 total number of instances. Table 1 describes the dataset used and Table 2 gives the information about the attributes of the table.

Table 1: Description of Dataset

Data Set Characteristics	*Multivariate*
Attribute Characteristics	*Integer, Categorical*
Associated Tasks	*Classification*
Number Of Instances	*284*
Number Of Attributes	*16*
Missing Values	*No*
Area	*Life*

Table 2: Description of Attributes

ATTRIBUTE	***DESCRIPTION***
Gender	*M(male), F(female)*
Age	*Age of the patient*
Smoking	✓ , ✗
Yellow Fingers	✓ , ✗
Anxiety	✓ , ✗
Peer Pressure	✓ , ✗
Chronic Disease	✓ , ✗
Fatigue	✓ , ✗
Allergy	✓ , ✗
Wheezing	✓ , ✗
Alcohol	✓ , ✗
Coughing	✓ , ✗
Shortness Of Breath	✓ , ✗
Swallowing Difficulty	✓ , ✗
Chest Pain	✓ , ✗
Lung Cancer	✓ , ✗

2. **Data Pre processing:** After adding the dataset in the canvas of azure ml studio, pre-processing of the data can be done using the available components present in the toolbox.

- **Clean Missing Values**: The "Clean Missing Values" module was used to substitute missing values with 0. The settings are shown in table 3. The fig 2 displays the output of the cleaned dataset.

Table 3: Clean Missing Values module settings

MODULE PARAMETER	SETTING
Selected Columns	all columns
Minimum Missing Value Ratio	0
Maximum Missing Value Ratio	1
Cleaning Mode	custom substitution value
Replacement Value	0

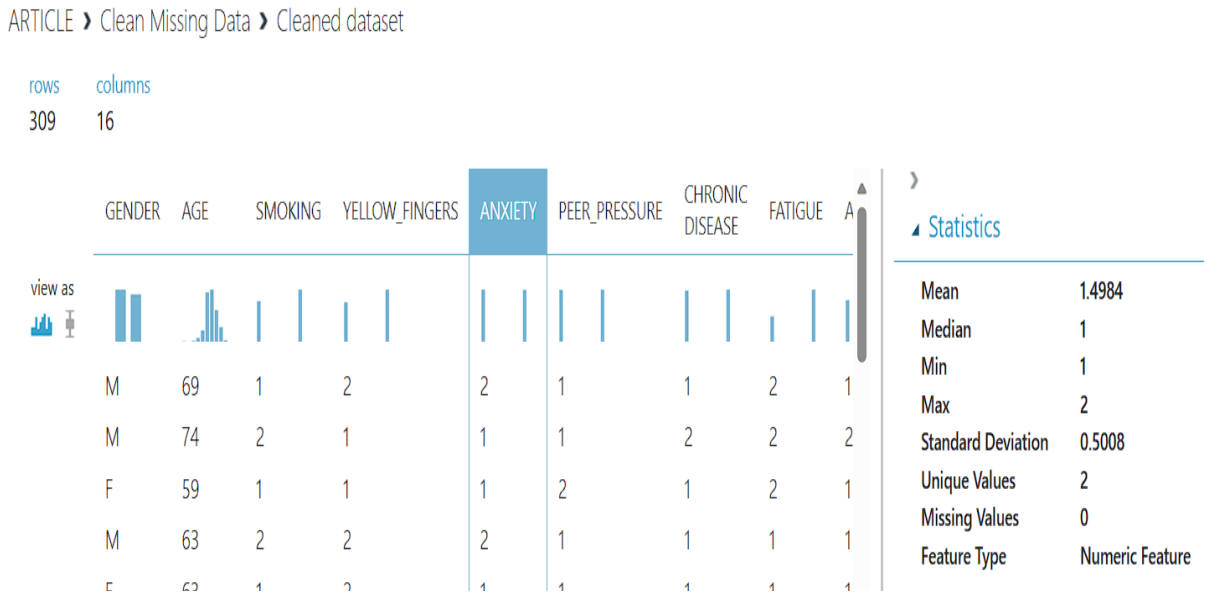


Figure 2: Output of clean missing value module

- **Feature Selection:** Feature selection was performed using the "Filter Based Feature Selection" module, which is discussed in detail in section 4.3 with scoring values and outputs. It is used to identify and select the most relevant and informative features. This helped to ignore redundant features and reduce the dimensionality of the dataset.

- **Splitting the Data Into Train and Test Sets:** The "Split Data" module was used to split the data into train and test sets before passing it to the model for training. Table 4 shows the module settings. Fig 3 shows the output of split dataset into train and test parts.

Table 4: Split Data Module Settings

MODULE PARAMETER	SETTING
Splitting mode	Split Rows
Fraction of rows in the first output dataset	0.7
Randomized split	True
Random seed	0
Stratified split	False

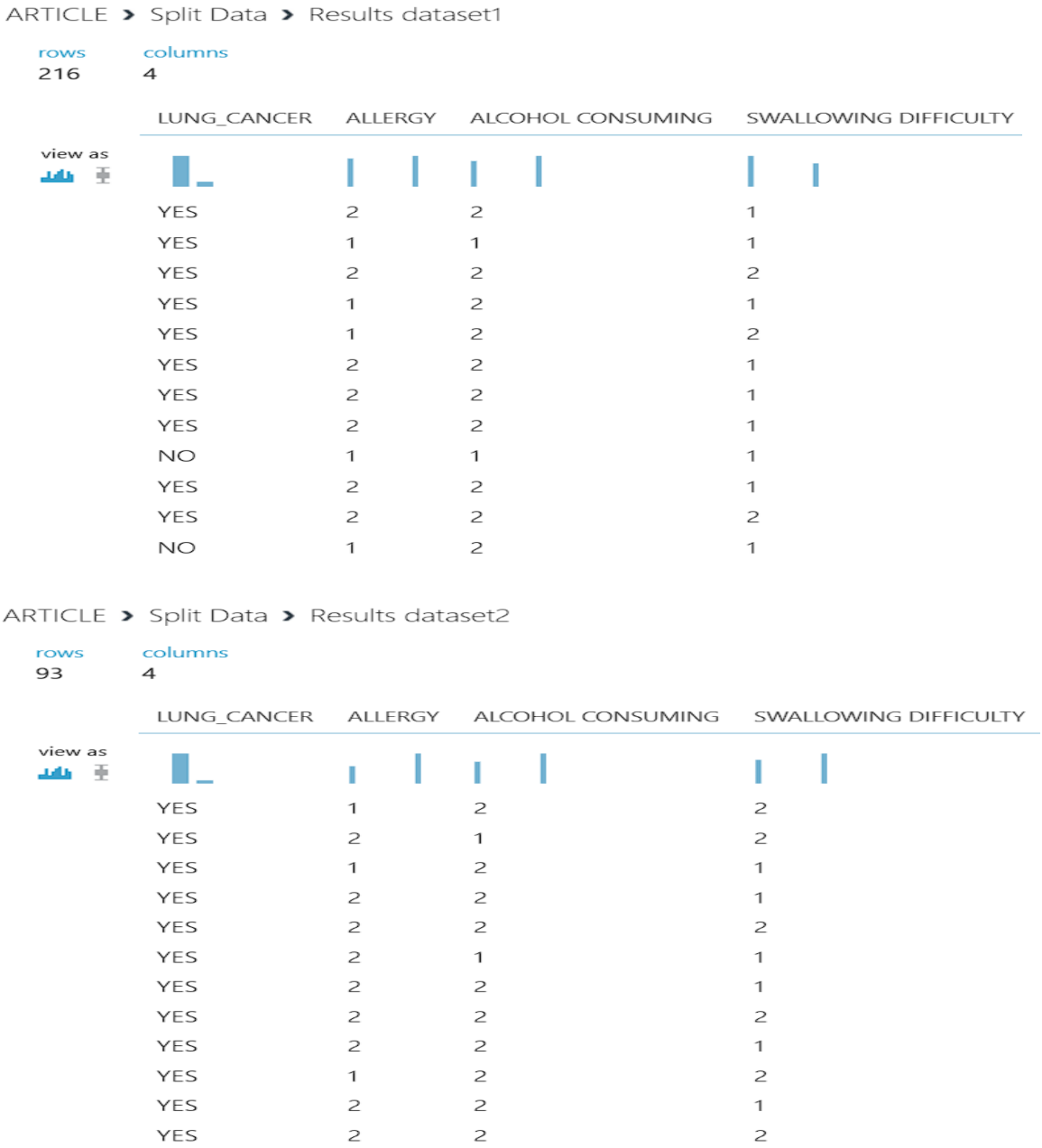


Figure 3: Output of Split Data module

The following pre-processing steps were performed specifically for the neural network model:

- **Normalization:** Data normalisation was done using the "Normalise Data" module. For neural networks, this is a typical pre-processing step. For normalisation, the Z-score approach has been utilised. A typical method of normalisation is called z-score normalisation, which includes dividing each feature's standard deviation by its value after deducting the mean of the feature from its value. Thus, a new feature is produced with a mean of 0 and a standard deviation of 1.

$$Z = \frac{x - \mu}{\sigma} \quad \textbf{(1)}$$

Where,

Z = standard score

x = observed value

μ = mean of the sample

σ = standard deviation of the sample

Table 5 depicts the settings of normalization module.

Table 5: Normalization Module Settings

MODULE PARAMETER	SETTING
Transformation Method	Z Score
Use 0 for Constant Columns when checked	True
Columns to Transform	Numeric, All

When all the values were initially positive, it can be seen in fig 4 that they have turned negative. Furthermore, even if the original values were all positive, it is very usual for the z-score normalised values to be negative. This is because the distribution of the data is considered during the z-score normalisation procedure, and if the distribution is not exactly symmetrical, negative values may result.

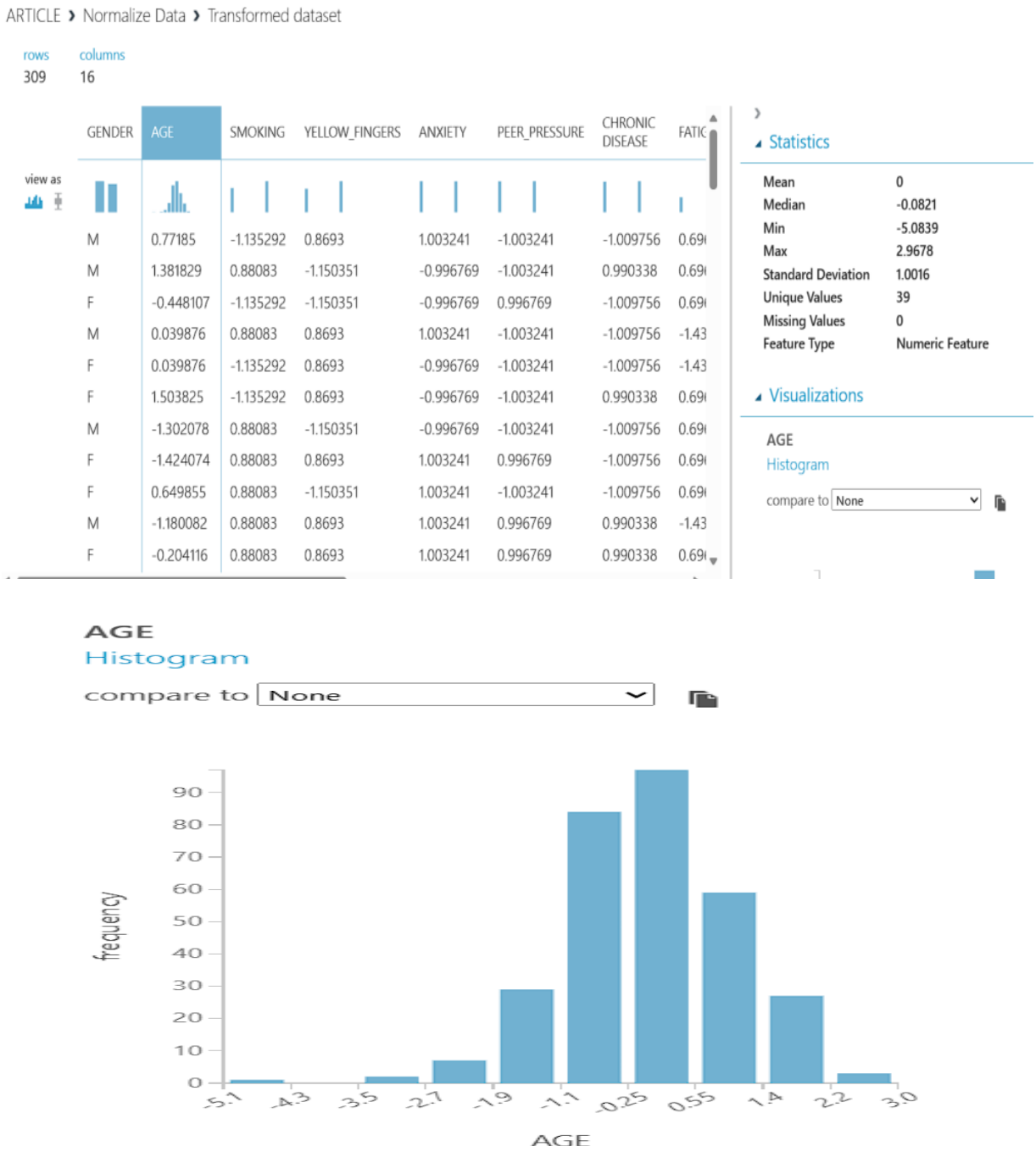


Figure 4: Output of normalization module

- **Principal component analysis (PCA):** The "Principal Component Analysis" module was used to reduce the dimensionality of the data to three principal components. This can improve the performance of neural networks by making them less complex and less prone to overfitting. PCA was applied to all columns except the target feature, "LUNG_CANCER." The table 6 shows the module settings and fig 5 shows the result of PCA.

Table 6: Principal Component Analysis Module Settings

MODULE PARAMETER	SETTING
Column Names	ALLERGY, WHEEZING, ALCOHOL CONSUMING, COUGHING, SWALLOWING DIFFICULTY, GENDER, AGE, SMOKING, YELLOW_FINGERS, ANXIETY, PEER_PRESSURE, CHRONIC DISEASE, FATIGUE, SHORTNESS OF BREATH, CHEST PAIN
Number of Dimensions to Reduce to	3
Normalize Dense Columns of Dataset to Zero Mean	TRUE

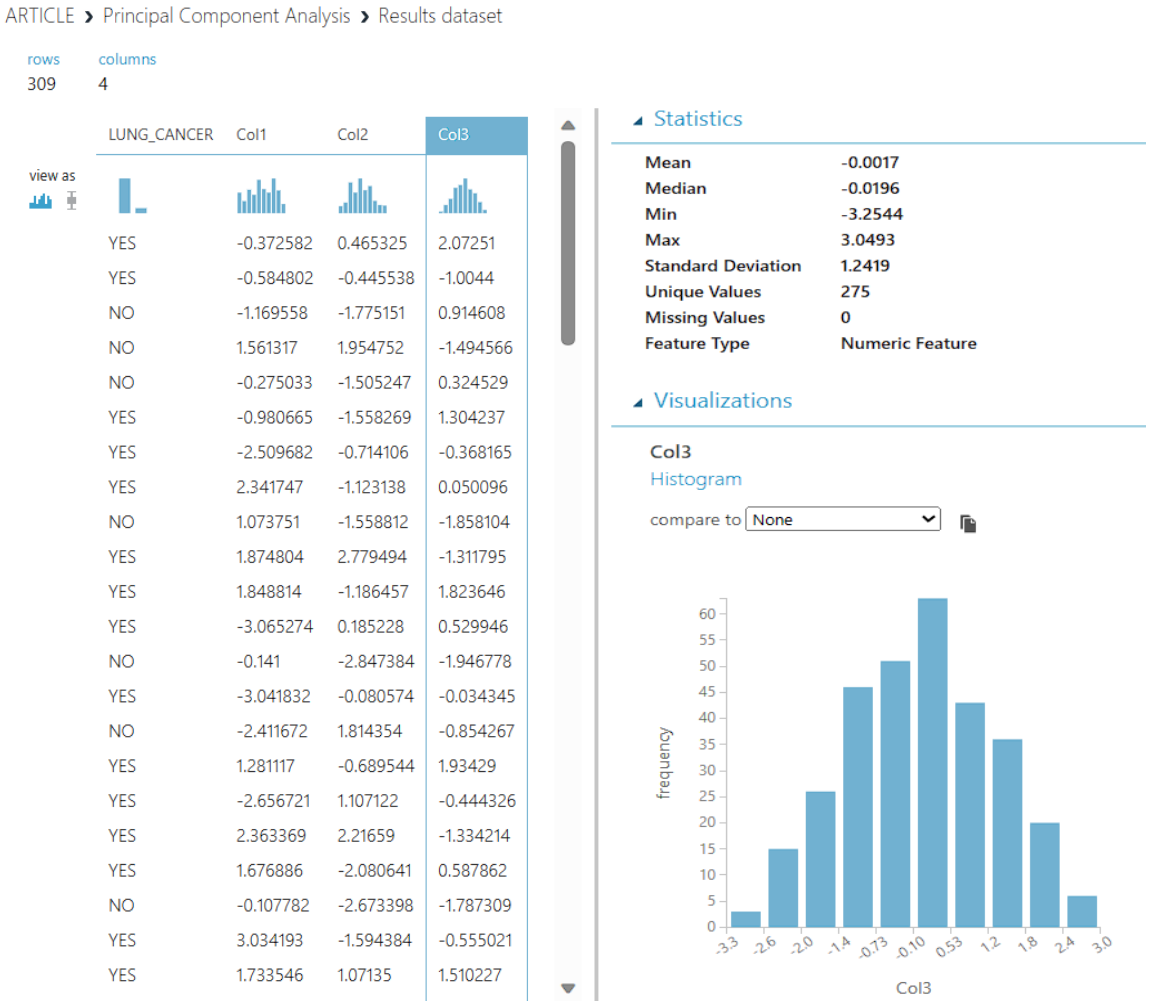


Figure 5: Result of PCA

Because they can enhance the performance, convergence speed, and interpretability of neural network models, normalisation and PCA are frequently utilised in neural network applications.

The neural network weights are less likely to become stuck in local minima as a result of normalisation, which can aid to accelerate training's convergence. A neural network's parameter count can be decreased via PCA, which makes it simpler to train and less prone to over fitting. By removing redundant features and lowering the dimensionality of the data, PCA can help increase the interpretability of a neural network.

3. **Feature Selection:** Feature selection is a method for reducing the number of features in an image by deleting unimportant, redundant, or noisy features from the image. Better learning performance, i.e., higher accuracy, less expensive computation, and easier model interpretation, might result from feature selection. A variety of feature selection algorithms have recently been presented by researchers in the fields of computer vision, text mining, and other fields. They have demonstrated the effectiveness of their works through theory and experiment [15].For feature selection step, filter-based feature selection technique is employed for the chosen models, which is discussed below:

4. **Filter Based Feature Selection:** Feature based filter selection is the module in Azure Machine Learning Studio, which offers various correlation coefficients. In a correlation analysis, the correlation coefficient is the sensitive indicator that evaluates the degree of the linear relationship between two variables.

The "Filter Based Feature Selection" module is used with the following parameter settings exhibited in table 7:

Table 7: Filter Based Feature Selection Module Settings

MODULE PARAMETER	SETTING
Column Names	LUNG_CANCER
Number of Desired Features	5

Different coefficients of correlation are discussed below:

- **Pearson Correlation Coefficient:** Pearson correlation coefficient is a statistical test used to figure out the statistical association between two continuous variables. It is the best method for quantifying the relationship between variables of importance because it is based on the covariance method. It denotes the magnitude of the correlation, as well as the direction of the relationship [16]. In this work, the Pearson coefficient of correlation was used to train the model. Equation for Pearson Correlation Coefficient is,

$$\mathbf{r} = \frac{\mathbf{n}(\sum \mathbf{xy}) - (\sum \mathbf{x})(\sum \mathbf{y})}{\sqrt{[\mathbf{n}\sum \mathbf{x^2} - (\sum \mathbf{x})^2][\mathbf{n}\sum \mathbf{y^2} - (\sum \mathbf{y})^2]}} \quad \textbf{(2)}$$

where,

r = Pearson Coefficient
n = number of the pairs of the stock
$\sum xy$ = sum of products of the paired stocks
$\sum x$ = sum of the x scores
$\sum y$= sum of the y scores
$\sum x^2$ = sum of the squared x scores
$\sum y^2$ = sum of the squared y scores

Table 8 shows the scoring values of Pearson correlation and fig 6 illustrates the output of filtered dataset.

Table 8: Feature Scoring Values from Pearson Correlation

Pearson	**Features**
1	Lung Cancer
0.32777	Allergy
0.28853	Alcohol Consuming
0.25973	Swallowing Difficulty
0.2493	Wheezing
0.24857	Coughing
0.19045	Chest Pain
0.18639	Peer Pressure
0.18134	Yellow Fingers
0.15067	Fatigue
0.14495	Anxiety
0.11089	Chronic Disease
0.08947	Age
0.060738	Shortness Of Breath
0.05818	Smoking
0	Gender

LUNG_CANCER	ALLERGY	ALCOHOL CONSUMING	SWALLOWING DIFFICULTY	WHEEZING	COUGHING
YES	1	2	2	2	2
YES	2	1	2	1	1
NO	1	1	1	2	2
NO	1	2	2	1	1
NO	1	1	1	2	2
YES	2	1	1	2	2

Figure 6: Output of filtered dataset after Pearson correlation

- **Chi Squared Correlation Coefficient:** By studying the relationship between the characteristics, the chi-square test aids in feature selection. It is sensitive to lower frequencies in table cells. In general, when the predicted value in a table cell is smaller than 5, chi-square can lead to incorrect results. Correlation tests can be used to choose features in machine learning. A chi-squared test can be used to determine whether the input variables are relevant to the output variable in classification issues where the output variable is categorical, and the input variables are similarly categorical. [17]

$$\chi^2 = \sum \frac{(O_i - E_i)^2}{E_i} \quad \textbf{(3)}$$

where,

O_i = observed value (actual value)

E_i = expected value

Feature scoring values generated can be observed in table 9 and filtered dataset is represented in fig 7.

Table 9: Feature Scoring Values from Chi Squared Correlation

Chi Squared	**Features**
1	Lung Cancer
33.196	Allergy
25.7246	Alcohol Consuming
20.845	Swallowing Difficulty
19.2045	Wheezing
19.0922	Coughing
12.167	Age
11.2079	Chest Pain
10.7348	Peer Pressure
10.1611	Yellow Fingers
7.01502	Fatigue
6.49199	Anxiety
3.79972	Chronic Disease
1.397645	Gender
1.13995	Shortness Of Breath
1.0459	Smoking

LUNG_CANCER	ALLERGY	ALCOHOL CONSUMING	SWALLOWING DIFFICULTY	WHEEZING	COUGHING
YES	1	2	2	2	2
YES	2	1	2	1	1
NO	1	1	1	2	2
NO	1	2	2	1	1
NO	1	1	1	2	2
YES	2	1	1	2	2

Figure 7: Output of filtered dataset after Chi Squared correlation

- **Kendall Correlation Coefficient:** When data is ranked by quantity, Kendall rank correlation is employed to investigate for similarities. Kendall's correlation coefficient uses pairs of data to determine the strength of link according to the pattern of concordance and discordance between the pairings [18]. Kendall's is frequently employed when data does not meet one of Pearson correlation conditions. Kendall's method is non-parametric, which means it does not demand for the two variables to follow a bell curve. It is defined as,

$$\tau = \frac{n_c - n_d}{n(n-1)/2} \quad (4)$$

Where,
n_c =no. of concordant pairs
n_d = no. of discordant pairs

Table 10 gives the scoring values of Kendall correlation coefficient while fig 8 depicts the output of the filtered dataset.

LUNG_CANCER	ALLERGY	ALCOHOL CONSUMING	SWALLOWING DIFFICULTY	WHEEZING	COUGHING
YES	1	2	2	2	2
YES	2	1	2	1	1
NO	1	1	1	2	2
NO	1	2	2	1	1
NO	1	1	1	2	2
YES	2	1	1	2	2

Figure 8: Output of filtered dataset after Kendall correlation

Table 10: Feature Scoring Values from Kendall Correlation

Kendall	Features
1	Lung Cancer
0.32777	Allergy
0.28853	Alcohol Consuming
0.25973	Swallowing Difficulty
0.2493	Wheezing
0.24857	Coughing
0.19045	Chest Pain
0.18639	Peer Pressure
0.18134	Yellow Fingers
0.15067	Fatigue
0.14495	Anxiety
0.11089	Chronic Disease
0.06332	Age
0.060738	Shortness Of Breath
0.05818	Smoking
0	Gender

- **Spearman Correlation Coefficient:** Spearman's Correlation shows the degree and direction of the monotonic relationship between the given two variables [19]. It has the advantage of being easier to compute, although in a data science context, it is unlikely to be doing anything by hand, and both approaches are computationally light in comparison to many other jobs.

$$\rho = 1 - \frac{6 \sum d_i^2}{n(n^2 - 1)} \quad \textbf{(5)}$$

where,

ρ = Spearman's rank correlation coefficient
n = number of observations
d_i = difference between the two ranks of each observation

Spearman correlation scoring values is tabulated in table 11 and output of the selected features is displayed in fig 9.

Table 11: Feature Scoring Values from Spearman Correlation

Spearman	Features
1	Lung Cancer
0.32777	Allergy
0.28853	Alcohol Consuming
0.25973	Swallowing Difficulty
0.2493	Wheezing
0.24857	Coughing
0.19045	Chest Pain
0.18639	Peer Pressure
0.18134	Yellow Fingers
0.15067	Fatigue
0.14495	Anxiety
0.11089	Chronic Disease
0.07606	Age
0.060738	Shortness Of Breath
0.05818	Smoking
0	Gender

LUNG_CANCER	ALLERGY	ALCOHOL CONSUMING	SWALLOWING DIFFICULTY	WHEEZING	COUGHING
YES	1	2	2	2	2
YES	2	1	2	1	1
NO	1	1	1	2	2
NO	1	2	2	1	1
NO	1	1	1	2	2
YES	2	1	1	2	2

Figure 9: Output of filtered dataset after Spearman correlation

- **Fisher Score Correlation Coefficient:** Fisher score is a supervised feature selection method based on filters and feature weights. Fisher score models have several benefits connected to the application of supervised learning for selecting features, such as simplified calculations, improved accuracy, and enhanced operability, which will minimise time-space complexity [20]. Equation of Fisher Z-score is evaluated as,

$$\mathbf{Z_r = \frac{\ln\left(\frac{1+r}{1-r}\right)}{2}} \quad \textbf{(6)}$$

Where,
r = Pearson correlation coefficient
Z_r = Fisher Z transformation

LUNG_CANCER	ALLERGY	ALCOHOL CONSUMING	SWALLOWING DIFFICULTY	WHEEZING	COUGHING
YES	1	2	2	2	2
YES	2	1	2	1	1
NO	1	1	1	2	2
NO	1	2	2	1	1
NO	1	1	1	2	2
YES	2	1	1	2	2

Figure 10: Output of filtered dataset after Fisher correlation

Table 12 records the Fischer score of the features while fig 10 depicts the output after filtering.

Table 12: Feature scoring values from Fisher correlation

Fisher Score	Features
1	Lung Cancer
0.12036	Allergy
0.09081	Alcohol Consuming
0.07234	Swallowing Difficulty
0.06627	Wheezing
0.06586	Coughing
0.03764	Chest Pain
0.03599	Peer Pressure
0.034	Yellow Fingers
0.02323	Fatigue
0.02146	Anxiety
0.01245	Chronic Disease
0.00807	Age
0.0037	Shortness of Breath
0.0034	Smoking
0	Gender

As it can be observed from the scores and output of different feature scoring methods that the five most highly correlated features are tabulated in table 13:

Table 13: List of Highly Correlated Features

Highly Correlated Features
Allergy
Alcohol Consumption
Swallowing Difficulty
Wheezing
Coughing

These five features are selected for training of the models- Logistic Regression and Decision Tree whereas principal component analysis has been done for Neural Network.

5. Classifiers

1. Overview of Classifiers

- **Logistic Regression:** It is a supervised learning technique to calculate the likelihood of a binary event. A binary logistic regression has a dependent variable with two possible values: lose/draw, pass/fail, spam/not spam, true/false, and the like. Mathematical equation for logistic regression is,

$$P = \frac{1}{1 + e^{-(a+bx)}} \quad (7)$$

where,

P = Probability of a 1 (the proportion of 1s, the mean of Y)
e = Base of the natural logarithm (about 2.718)
a and b = Parameters of the model

An illustration of logistic regression is the use of computer learning to forecast whether a person will likely develop a cold. It is referred to as binary categorization since there are only two viable answers to this question: yes, they are infected, or no, they are not. Although logistic regression can occasionally be difficult, doing regression analysis is made simple by clever statistics tools [21]. It explains the relationship between a single dependent variable and one or more nominal independent variables.

Several underlying assumptions guide logistic regression. Binary logistic regression initially requires a binary dependent variable, but ordinal logistic regression just requires an ordinal dependent variable. Second, logistic regression requires independent observations. The observations should not be based on matched or repeated measurements, to put it another way. Third, for logistic regression, there should be little to no multicollinearity among the independent variables. This implies that the independent variables should not have a high degree of correlation.

To ensure the optimal performance and generalization of the logistic regression classifier, meticulous configuration is done of the following hyper parameters for the two-class logistic regression model trained on Azure ML Studio shown in table 14:

Table 14: Values of Hyper parameters for logistic regression model

Hyper parameter	**Value**
Create Trainer Mode	Single Parameter
Optimization Tolerance	1E-07
L1 Regularization Weight	1

L2 Regularization Weight	1
Memory Size for L-BFGS	20
Random Number Seed	None
Allow Unknown Categorical Levels	True
Quiet	True
Use Threads	True

- **Decision Tree:** It is a non-parametric supervised learning method that may be applied to both classification and regression problems. A decision tree is a diagram that shows the possible results of a number of connected decisions. It enables a person or organization to contrast alternative courses of action according to their costs, likelihoods, and returns. These can be used to start exploratory discussions or to create an algorithm that predicts the best choice analytically.

A root node, branches, internal nodes, and leaf nodes make up its hierarchical tree structure. There are three different sorts of nodes in a decision tree: choice, chance, and end nodes. The decision nodes, which are squares and signify a decision that needs to be made, are present. The chance node displays numerous outcomes that are unknown and is represented by circles. Triangles serve as the end node representations and denote an outcome. Schematic diagram of the decision tree is shown in Fig 11.

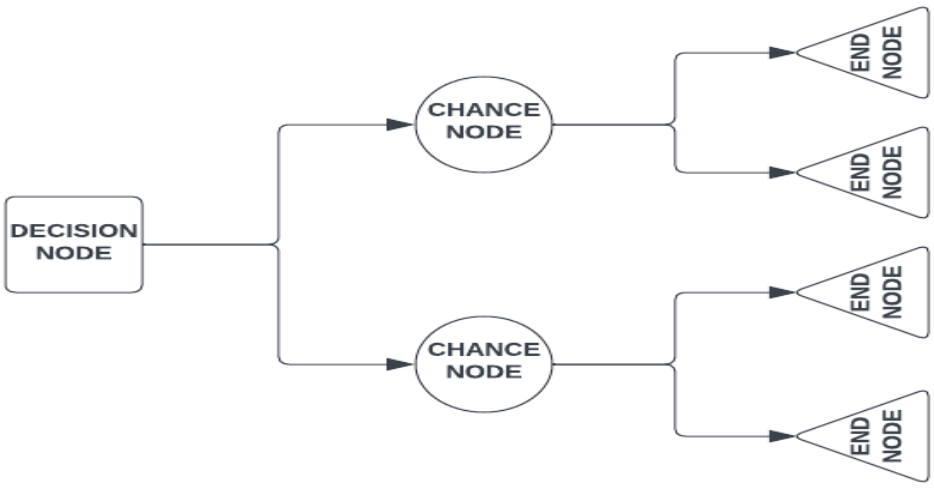


Figure 11: Decision Tree

A boosted decision tree is an ensemble learning approach. This encompasses that the successor trees correct the errors of all the predecessor trees cumulatively. Predictions thus depend on the ensemble of trees [22].

Boosted Decision Trees combine the strengths of decision trees and boosting, creating a robust ensemble model.

Decision trees are simple yet effective models that make sequential decisions based on input features. Each node represents a feature and a split based on a threshold, leading to a hierarchical structure. Decision trees partition data into regions, making them interpretable models.

Boosting is an ensemble technique that consolidates multiple weak learners into a strong learner. It does this iteratively by giving more weight to misclassified examples in each round, allowing the model to focus on challenging instances. The final prediction is a weighted sum of weak learners.

Mathematically, the ensemble prediction of a boosted decision tree model can be represented as:

$$F(x) = \sum_{t=1}^{T} \alpha_t h_t(x) \quad \textbf{(8)}$$

Where,
F(x) is the ensemble prediction for inputx.
α_t is the weight assigned to the classifier $h_t(x)$.
h_t (x) is the prediction of thet-th decision tree classifier for input x.
T is the total number of decision trees in the ensemble

To ensure optimal performance and generalization of the Two-Class Boosted Decision Tree classifier which is an example of a gradient boosting machine (GBM), meticulous configuration is done of the following hyper parameters for the two-class boosted decision tree model trained on Azure ML Studio shown in table 15:

Table 15: Values of Hyper parameters for Decision Tree Model

Hyper Parameter	**Value**
Create Trainer Mode	Single Parameter
Maximum Number of Leaves per Tree	20
Minimum Number of Samples per Leaf Node	10
Learning Rate	0.2
Number of Trees Constructed	100
Random Number Seed	None
Allow Unknown Categorical Levels	True

- **Neural Network:** The neural network classification approach is a supervised learning method.An input layer, one or more hidden layers, and an output layer are all components of neural networks. With own weight and threshold, each node is connected to the others [23]. If a node's output surpasses a certain threshold value, that node is triggered and begins sending data to the network's next layer. If the threshold is greater than the output of the node, no data is then forwarded to the next network layer.

A neural network is a collection of algorithms with the objective to discover the fundamental relationships among a collection of data by employing a similar procedure to that of human brain follows [24]. Fig 12 shows the architecture of the neural network.

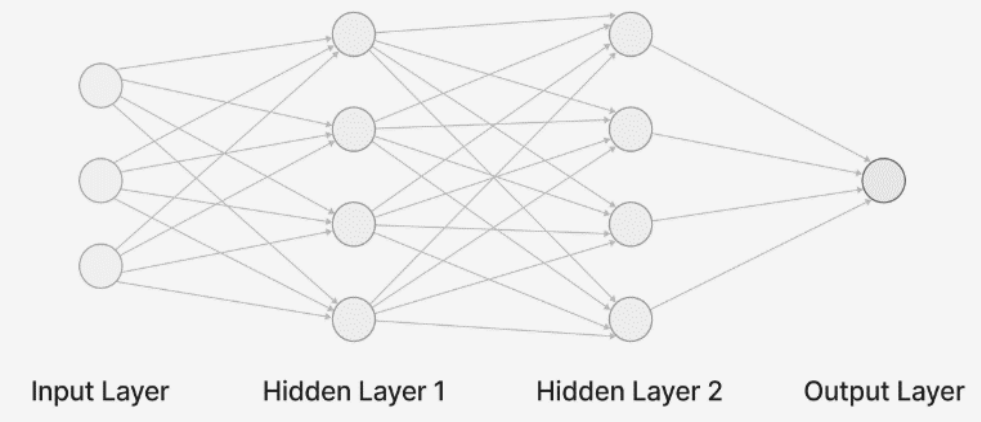


Figure 12: Schematic diagram of Neural Network

To improve the model's performance, Principal Component Analysis (PCA) is applied. Being a statistical method, it summarises the information in large data tables by employing a smaller set of "summary indices" that could be more easily shown and analysed. PCA is used for dimensionality reduction.

The Two-Class Neural Network is designed to predict binary outcomes by learning complex relationships within the data. It utilizes activation functions, weight parameters, and a loss function (in this case, Cross Entropy) to optimize its predictions.

To ensure the optimal performance of the Two-Class Neural Network classifier, meticulous configuration is done of the following hyper parametersfor the binary neural network component trained on Azure ML studioshown in table 16:

Table 16: Values of Hyper Parameters for Neural Network Model

Hyper Parameter	**Value**
Create Trainer Mode	Single Parameter
Hidden Layer Specification	Fully connected case
Loss Function	Cross Entropy
Number of Hidden Layers	1
Number of Hidden Nodes	100
Batch Size	32
Number of Epochs	10
Learning Rate	0.1
Number of Learning Iterations	100

Initial Learning Weights Diameter	0.1
Momentum	0
Type of Normalizer	Min-Max Normalizer
Shuffle Examples	True
Random Number Seed	None
Activation Function	Sigmoid
Allow Unknown Categorical Levels	True

For early detection of lung cancer, machine learning algorithms are implemented in the experiment: Binary Logistic Regression, Binary Boosted Decision Tree, and Binary Neural Network. For better accuracy, feature selection is also performed prior to model training.

6. **Hyper Parameter Tuning of Classifiers:** The hyper parameter tuning process for the Logistic Regression, Boosted Decision Tree and Neural Network models was conducted methodically, considering the specified hyper parameters introduced in tables14, 15 &16. The goal was to maximize model performance while considering that the default values set in Azure ML Studio were already providing satisfactory results with minor adjustments. The process involved the following steps:

- **Initialization:**
  - **Logistic Regression:** Initialize a logistic regression model using default hyperparameter settings provided by Azure ML Studio.
  - **Decision Trees:** Initialize a Two-Class Boosted Decision Tree model using default hyperparameter settings.
  - **Neural Network:** Initialize a Two-Class Neural Network model using default hyperparameter settings.

- **Performance Metric Selection:**
  - Evaluate model performance using commonly used metrics such as AUC, accuracy, and F1 score.
  - Note that default settings already yielded satisfactory results for all models.

- **Hyperparameter Exploration:**
  - **Logistic Regression:**
    - Considered hyperparameters such as 'Create Trainer Mode,' 'Optimization Tolerance,' 'L1 Regularization Weight,' 'L2 Regularization Weight,' etc.
    - Default values offered a reasonable trade-off between model complexity and performance.
  - **Decision Trees:**
    - Considered hyperparameters such as 'Create Trainer Mode,' 'Maximum Number of Leaves per Tree,' 'Minimum Number of Samples per Leaf Node,' etc.
    - Default values offered a reasonable trade-off between model complexity and performance.

- **Neural Network:**
  - Considered hyperparameters such as 'Create Trainer Mode,' 'Hidden Layer Specification,' 'Loss Function,' 'Number of Hidden Nodes,' etc.
  - Default values offered a reasonable trade-off between model complexity and performance.

- **Minor Tinkering:**
  - For each model, made minor adjustments to hyperparameters where necessary to maximize performance.
  - Observed that these adjustments had only a negligible impact on the model's performance metrics, reaffirming the effectiveness of the default configurations.

- **Optimal Default Configuration:**
  - Given the minimal changes in performance metrics resulting from hyperparameter adjustments for all three models, the default configurations provided by Azure ML Studio were considered optimal for their respective classification tasks.

7. **Data Balancing Strategies:** In many real-world scenarios, imbalanced datasets are quite common. This imbalance can occur due to various factors, including the nature of the problem, data collection process, or inherent characteristics of the target variable. Imbalanced datasets can be found in domains such as fraud detection, medical diagnosis, rare event prediction, and more.

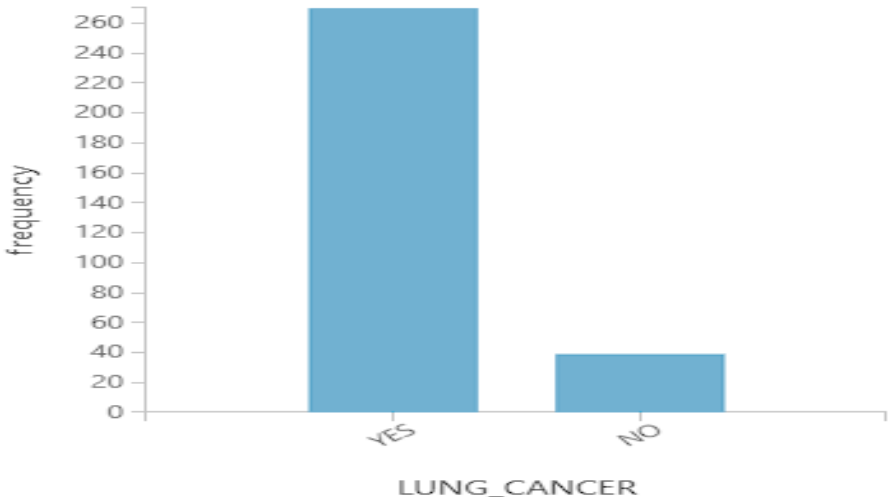


Figure 13: Visualization of target variable “LUNG_CANCER” of the dataset

The lung cancer dataset exhibits a clear class imbalance, which is readily apparent from the visualization of the target feature in Fig 13. The imbalance is evident in the significant disparity between the two classes, with one class being significantly more prevalent than the other. This imbalance in class distribution underscores the importance of selecting appropriate modelling techniques and strategies to address this challenge effectively.

- **Regularization (Logistic Regression):**
  - **Over Fitting Mitigation:** Regularization techniques like L1 (Lasso) or L2 (Ridge) can help prevent over fitting, which is especially important when dealing with limited data in the minority class.
  - **Bias Reduction:** Regularization can reduce the model's bias towards the majority class, making it more balanced in its predictions.

- **Boosted Decision Trees (GBM):**
  - **Ensemble Strength:** Boosting techniques combine multiple decision trees, each focusing on different aspects of the data. This ensemble approach can better capture the complexity of imbalanced data and make better use of the limited minority class samples.
  - **Adaptation to Class Imbalance:** Boosting algorithms assign higher weights to misclassified instances, giving more attention to the minority class. This adaptability makes them well-suited for imbalanced datasets.

- **Neural Network:**
  - **Feature Representation Learning:** Neural networks can automatically learn informative features from raw data, potentially helping the model identify subtle patterns in the minority class.
  - **Complexity Handling:** Deep neural networks can capture intricate relationships between features, which can be essential for addressing class imbalance.
  - **Resilience to Imbalance:** Neural networks, especially with proper architecture and hyperparameter tuning, can adapt to class imbalance by learning complex decision boundaries.

When evaluating these models on imbalanced data, it is important to use appropriate metrics [25] like precision, recall, F1-score, AUC-ROC, or AUC-PR. These metrics provide a more comprehensive view of model performance than accuracy alone, as they consider the trade-off between false positives and false negatives, which is crucial when dealing with imbalanced datasets.

The presence of an imbalanced dataset is common in many practical applications, and the choice to train models like Logistic Regression with regularization, Boosted Decision Trees, and Neural Networks is justified due to their abilities to mitigate the challenges associated with class imbalance. By carefully selecting and tuning these models and using appropriate evaluation metrics, it is possible to achieve good results and make meaningful predictions even in the presence of imbalanced data.

8. **Justification for the Selection of Classifiers:** In this research chapter, the choice of classifiers for lung cancer detection within an imbalanced dataset is justified. Logistic Regression with Regularization, Gradient Boosting Machine, and Neural Networks in Azure ML Studio have been selected. These selections align with the research objectives, offering distinct advantages to address the complexities of the dataset and contribute to the research on lung cancer detection.

- **Logistic Regression with Regularization:**
  - **Interpretability:** Logistic Regression offers interpretability, vital in healthcare, allowing clinicians to understand the impact of risk factors.
  - **Handling Imbalance:** Regularized Logistic Regression manages class imbalance effectively, preventing over fitting and ensuring model stability.
  - **Efficiency:** It is computationally efficient and seamlessly integrated into Azure ML Studio, making it practical.
- **Boosted Decision Trees (Gradient Boosting Machine):**
  - **Non-Linearity and Feature Importance:** Boosted Decision Trees capture complex relationships and rank feature importance, aiding in identifying risk factors.
  - **Imbalance Adaptation:** They naturally adapt to class imbalance, focusing on predicting rare events accurately.
  - **Integration:** Azure ML Studio provides seamless integration with Gradient Boosting Machine algorithms, ensuring ease of use.
- **Neural Networks:**
  - **Complex Feature Extraction:** Neural Networks, particularly deep ones, automatically extract complex and hierarchical features, vital for capturing subtle patterns in lung cancer data.
  - **Multi-Modal Data:** They seamlessly integrate diverse data types, such as patient history, images, and genomic information, crucial in healthcare.
  - **Availability:** Azure ML Studio offers neural network frameworks, making it convenient for deep learning model development and fine-tuning.

9. Experiment Procedure

- **Sub Experiment 1 – using Linear Regression:** After data pre-processing, for classification of lung cancer, two class logistic regression have been performed. During the feature selection Pearson correlation featuring scoring method is used with five number of desired features, refer to table 13 for the list of highly correlated features with the target attribute being “LUNG_CANCER”.

  After feature selection, 70% of data is used for training of the model and 30% of data is taken for testing of the model. After training the model for logistic regression, score model module of the environment is used for scoring the trained model for classification or regression. ‘Evaluate model’ module of the environment is then used for evaluating the trained model shown in fig 14.

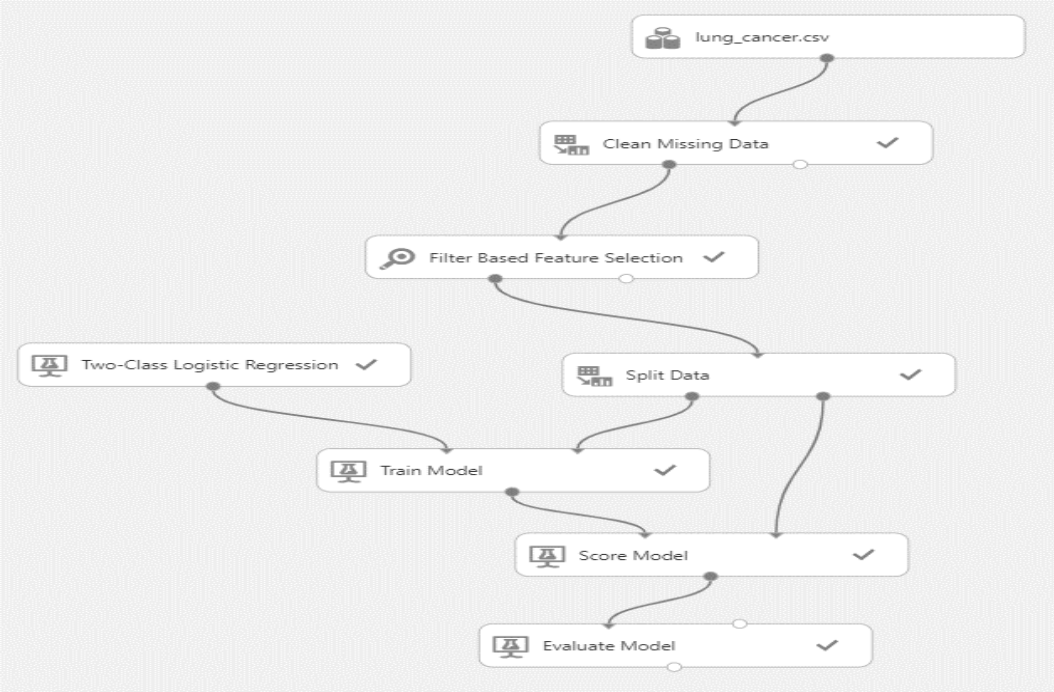


Figure 14: Schematic view of Logistic Regression Model in Azure ML Studio

- **Sub Experiment 2 – using Boosted Decision Tree:** The final score of the model is recorded and results for the boosted decision tree algorithm are determined using a similar approach employing two class boosted decision tree modules displayed in fig 15.

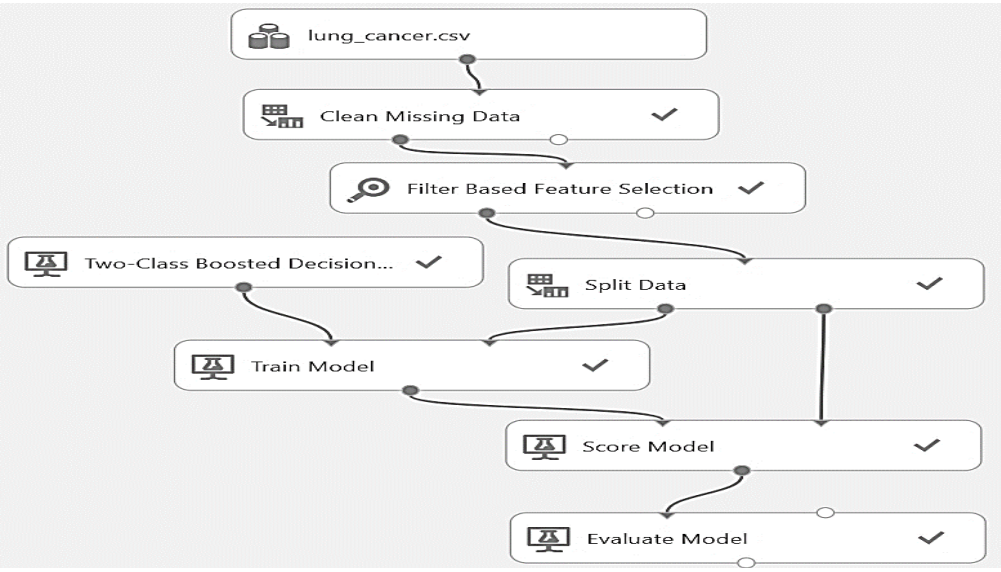


Figure 15: Schematic view of Boosted Decision Tree Model in Azure ML Studio

- **Sub Experiment 3 – using Neural Network:** For the training of model using two class neural network module, firstly normalization of the data is done using Z-Score transformation method. The Z-score is a statistical measurement that quantifies a

value's relationship to the mean of a set of values. After this PCA is implemented with three as number of dimension reduction. After that, the score of the model is calculated and the model is evaluated. The pipeline structure can be observed in fig 16.Also, an assorted flowchart for the entire experiment is portrayed in fig 17.

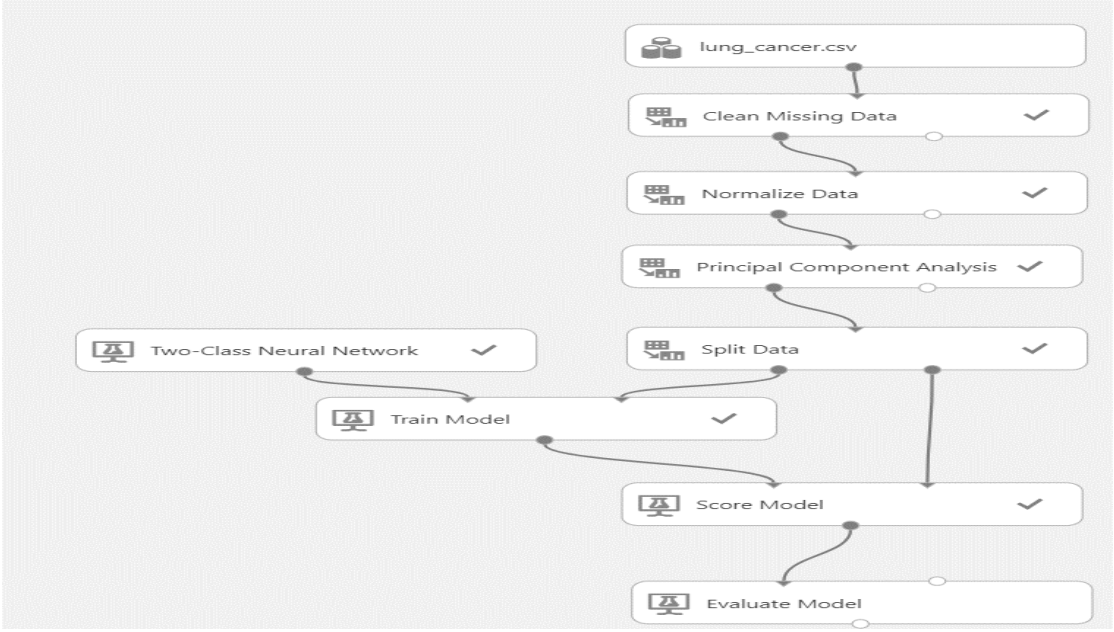


Figure 16: Schematic view of Neural Network Model in Azure ML Studio

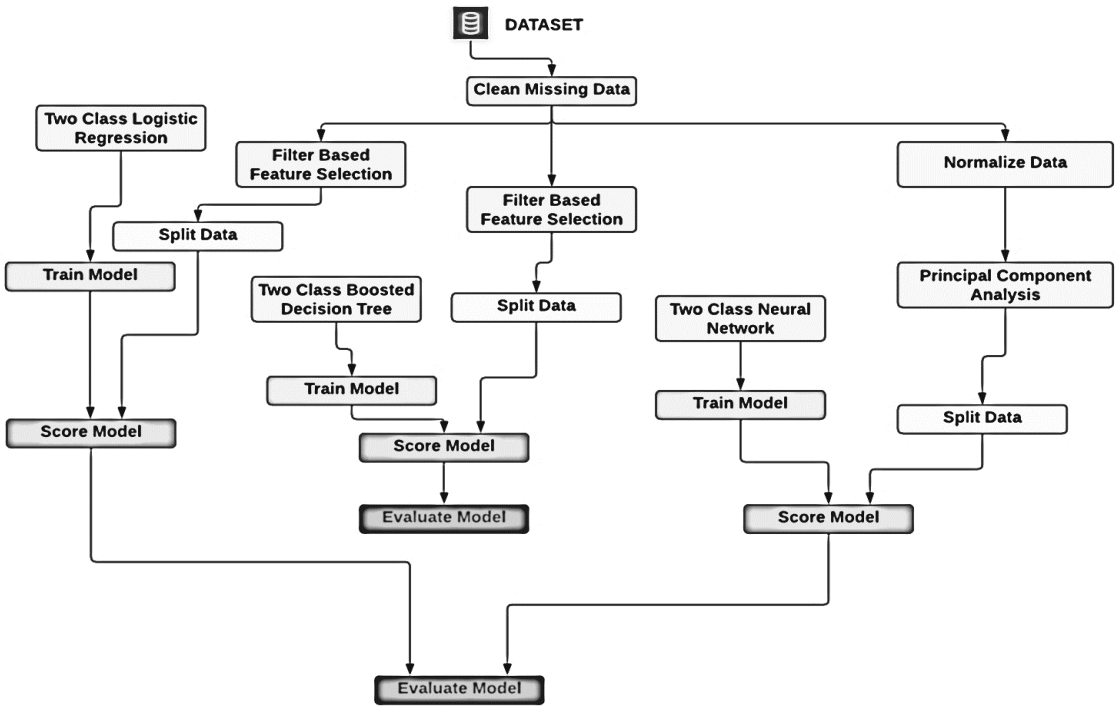


Figure 17: shows the flowchart that outlines the experiment explained with each Azure module representing a single step.

V. RESULTS AND DISCUSSION

Performance measures are used to evaluate the performance for ML models. The most common and easiest way to describe the performance of a classification problem is to form a confusion matrix.

Table 17: Confusion Matrices of the 3 Employed Classifiers

(a)

Predicted \ Actual	1	0
1	84 TP	9 FP
0	0 FN	0 TN

(a) Confusion matrix for logistic regression model

(b)

Predicted \ Actual	1	0
1	133 TP	18 FP
0	3 FN	0 TN

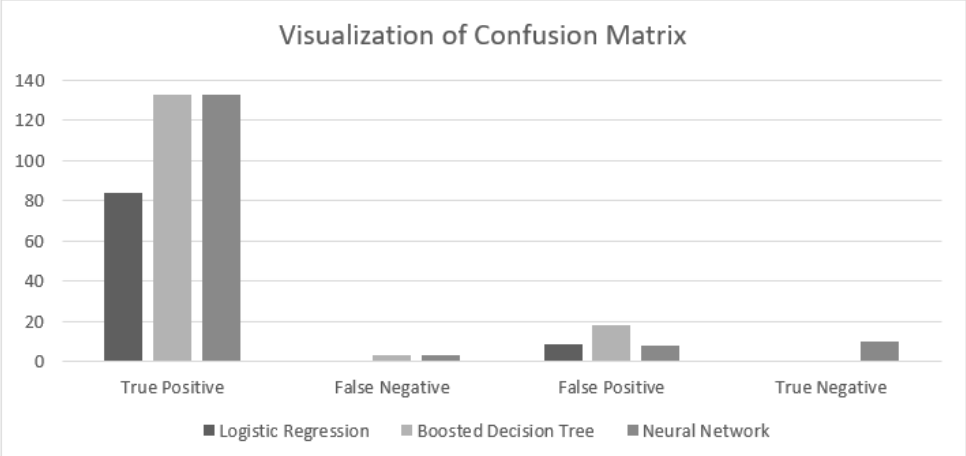
(b) Confusion matrixfor boosted decision tree model

(c)

Predicted \ Actual	1	0
1	*133* TP	8 FP
0	3 FN	10 TN

(c) Confusion matrix for neural network model

Graph 1: Shows A Visualization of Confusion Matrix in the Form of Column Graph

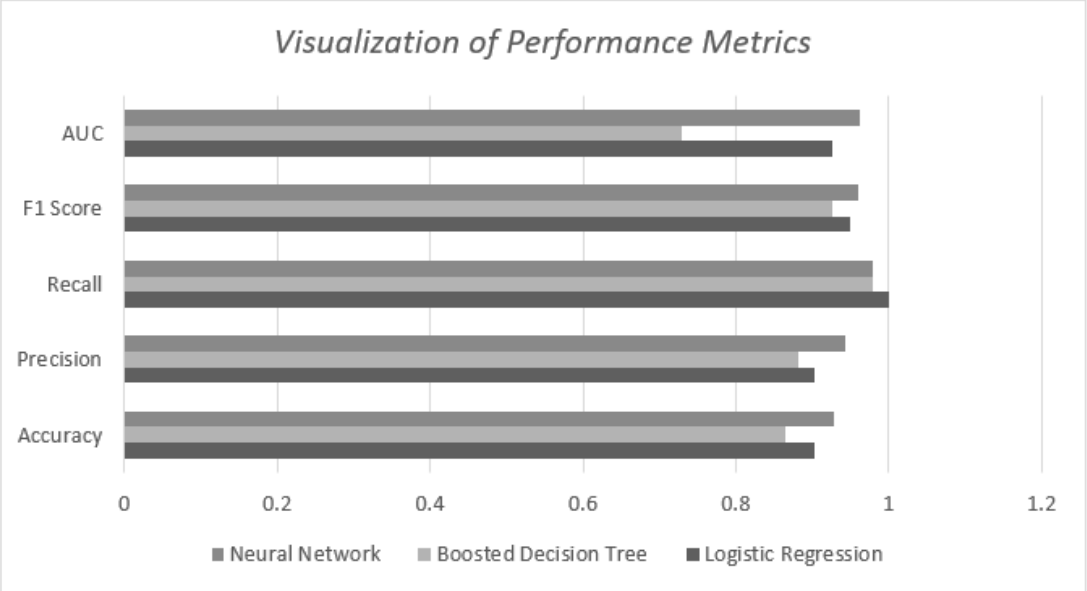


A classification report also considers performance metrics like accuracy precision, recall, F1 score and AUC.

Table 18: Evaluation of Classifiers on different Performance Measures

Performance Measures	Logistic Regression	Boosted Decision Tree	Neural Network
Accuracy	0.903	0.864	0.929
Precision	0.903	0.881	0.943
Recall	1.000	0.978	0.978
F1 Score	0.949	0.927	0.960
AUC	0.925	0.730	0.963
Positive Label	YES	YES	YES
Negative Label	NO	NO	NO

Graph 2: Shows a Visualization of Performance Metrics in the Form of Bar Graph



From the tables 17&18 and graphs1 & 2, it is evident that for this dataset, neural network is the optimal model having higher accuracy of 0.929, precision of 0.943 and F1 scoreof0.960 among the classifiers taken for the comparison study. Azure ML Studio provides a scalable environment with lots of customization, and as such, this experiment could further expand its scope by adding more classifiers for comparison or training with different feature set derived from some other correlation metrics.

VI. CONCLUSION

This study has examined the most recent developments in machine learning for the detection of lung cancer in this chapter. The various machine learning models that have been employed for this task, as well as the difficulties and potential benefits of doing so, have all been covered in this work. This study also included a case study of a machine learning model with an accuracy of 0.929 after being trained on a dataset of lung cancer patients. This is a substantial advancement over current lung cancer detection techniques like chest X-rays and CT scans, which typically have 70-80% accuracy levels. Overall, the findings of this research indicate that machine learning has the power to fundamentally alter how lung cancer is identified and treated. Lung cancer patients can be identified earlier and more precisely using machine learning algorithms to enhance patient outcomes. It is crucial to remember that every machine learning model is faultless. A model will still produce errors even if its accuracy is 0.929. To confirm lung cancer diagnosis, machine learning models should be used in concert with other diagnostic methods, such as biopsies. According to this research, creating machine learning models for lung cancer diagnosis is a considerable improvement. By assisting clinicians in making an earlier and more precise diagnosis of lung cancer, these models have the potential to save lives. This team intends to carry out more machine-learning research for lung cancer diagnosis in the future. Creating models that can identify lung cancer in its earliest stages, when it is most curable, is particularly relevant to this effort. This work also aims to create models that may be used to forecast the prognosis for patients with lung cancer to help clinicians choose the best course of therapy.

REFERENCES

[1] Y. Qiang, Y. Guo, X. Li, Q. Wang, H. Chen, and D. Cuic, "The diagnostic rules of peripheral lung cancer preliminary study based on data mining technique," Journal of Nanjing medical university, vol. 21, no. 3, pp. 190–195, 2007.

[2] KwetisheJoroDanjuma, "Performance Evaluation of Machine Learning Algorithms in Post-operative Life Expectancy in the Lung Cancer Patients" Department of Computer Science, ModibboAdama University of Technology, Yola, Adamawa State, Nigeria.

[3] Zehra Karhan1, Taner Tunç2, "Lung Cancer Detection and Classification with Classification Algorithms" IOSR Journal of Computer Engineering (IOSR-JCE) e-ISSN: 2278-0661,p-ISSN: 22788727, Volume 18, Issue 6, Ver. III (Nov.-Dec. 2016), PP 71-77.

[4] Ada, RajneetKaur, " A Study of Detection of Lung Cancer Using Data Mining Classification Techniques " International Journal of Advanced Research in Computer Science and Software Engineering, Volume 3, Issue 3, March 2013.

[5] L. Yu and H. Liu, "Feature selection for high-dimensional data: A fast correlation-based filter solution," in ICML, vol. 3, 2003, pp. 856–863.

[6] J. R. Quinlan, C4. 5: programs for machine learning. Elsevier, 2014.

[7] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten, "The weka data mining software: an update," ACM SIGKDD explorations newsletter, vol. 11, no. 1, pp. 10–18, 2009.

[8] F. Badjio and F. Poulet, "Dimension reduction for visual data mining," in international symposium on applied stochastic models and data analysis (ASMDA-2005), 2005.

[9] F. Badjio and F. Poulet, "Dimension reduction for visual data mining," in International symposium on applied stochastic models and data analysis (ASMDA-2005), 2005.

[10] E. Avci, "A new expert system for diagnosis of lung cancer: Gdalssvm," Journal of medical systems, vol. 36, no. 3, pp. 2005–2009, 2012.

[11] P. J. Tan and D. L. Dowe, "Mml inference of oblique decision trees," in AI 2004: Advances in Artificial Intelligence. Springer, 2005, pp. 1082–1088.

[12] S. M. Salaken, A. Khosravi, A. Khatami, S. Nahavandi and M. A. Hosen, "Lung cancer classification using deep learned features on low population dataset," 2017 IEEE 30th Canadian Conference on Electrical and Computer Engineering (CCECE), Windsor, ON, Canada, 2017, pp. 1-5, doi: 10.1109/CCECE.2017.7946700.

[13] Frogglew. (n.d.). What is azure machine learning? - azure machine learning. Azure Machine Learning | Microsoft Learn. Retrieved February 26, 2023, from https://learn.microsoft.com/en-us/azure/machine-learning/overview-what-is-azure-machine-learning

[14] Ahmad Bhat, M. (n.d.). Lung Cancer | Kaggle [Dataset]. https://www.kaggle.com/datasets/mysarahmadbhat/lung-cancer

[15] Jianyu Miao, Lingfeng Niu, A Survey on Feature Selection, Procedia Computer Science, Volume 91, 2016, Pages 919-926, ISSN 1877-0509,

[16] Haomiao Zhou, Zhihong Deng, Yuanqing Xia and Mengyin Fu, A new sampling method in particle filter based on Pearson correlation coefficient, Neurocomputing, http://dx.doi.org/10.1016/j.neucom.2016.07.036

[17] Plackett, R. L. (1983). Karl Pearson and the Chi-Squared Test. International Statistical Review / Revue Internationale de Statistique, 51(1), 59–72. https://doi.org/10.2307/1402731

[18] O'Gorman, T. W., & Woolson, R. F. (1995). Using Kendall's τb Correlations to Improve Variable Selection Methods in Case-Control Studies. Biometrics, 51(4), 1451–1460. https://doi.org/10.2307/2533275

[19] Schober, Patrick MD, PhD, MMedStat; Boer, Christa PhD, MSc; Schwarte, Lothar A. MD, PhD, MBA. Correlation Coefficients: Appropriate Use and Interpretation. Anesthesia& Analgesia 126(5):p 1763-1768, May 2018. | DOI: 10.1213/ANE.0000000000002864

[20] Lin Sun, Tianxiang Wang, Weiping Ding, Jiucheng Xu, Yaojin Lin, Feature selection using Fisher score and multilabel neighborhood rough sets for multilabel classification, Information Sciences, Volume 578, 2021, Pages 887-912, ISSN 0020-0255,https://doi.org/10.1016/j.ins.2021.08.032.

[21] R. Goyal, P. Maity, M. Bhatia and A. Grover, "Detecting Keratoconus using Machine Learning Models," 2022 4th International Conference on Artificial Intelligence and Speech Technology (AIST), Delhi, India, 2022, pp. 1-5, doi: 10.1109/AIST55798.2022.10065321.

[22] Likebupt. (n.d.). Two-class boosted decision tree: Component reference - azure machine learning. Two-Class Boosted Decision Tree: Component Reference - Azure Machine Learning | Microsoft Learn.

Retrieved February 26, 2023, from https://learn.microsoft.com/en-us/azure/machine-learning/component-reference/two-class-boosted-decision-tree

[23] M. Bhatia, S. Dhir, P. Tanwar and A. Khan, "Semantic Similarity based measurement for Lung's infection imagery using Deep Learning," 2022 12th International Conference on Cloud Computing, Data Science & Engineering (Confluence), Noida, India, 2022, pp. 129-133, doi: 10.1109/Confluence52989.2022.9734137.

[24] Shivam and Bareja, Pawan and Bhatia, Madhulika, Appearance of Machine and Deep Learning in Image Processing: A Portrayal (2018). International Journal of Computational Intelligence & IoT, Vol. 2, No. 4, 2018, Available at SSRN: https://ssrn.com/abstract=3361144

[25] D. K. Singh, Hitesh, V. Kumar, and H. Pham, "Decision Support System for Ranking of Software Reliability Growth Models," Springer Series in Reliability Engineering. Springer International Publishing, pp. 227–244, 2023. doi: 10.1007/978-3-031-21232-1_12.

