COMPARATIVE ANALYSIS OF ALGORITHMS FOR LUNG CANCER PREDICTION

A COURSE PROJECT REPORT

By

ASSVLY YASWANTH

(RA2111026010121)

VAMSI KRISHNA

(RA2111026010114)

Under the guidance of

DR.M.FERNI UKRIT
(Associate Professor)

In partial fulfillment for the Course

of

18CSE479T - STATISTICAL

MACHINE LEARNING

FACULTY OF ENGINEERING AND TECHNOLOGY

SRM INSTITUTE OF SCIENCE AND TECHNOLOGY

Kattankulathur, Chengalpattu District

NOVEMBER 2023

SRM INSTITUTE OF SCIENCE AND TECHNOLOGY

(Under Section 3 of UGC Act, 1956)

BONAFIDE CERTIFICATE

Certified that this mini project report "COMPARATIVE ANALYSIS OF ALGORITHMS FOR "LUNG CANCER PREDICTION" is the bonafide work of **ASSVLY YASHWANTH (RA2111026010121), VAMSI KRISHNA (RA2111026010114)** who carried out the project work under my supervision.

SIGNATURE

Dr.M.Ferni Ukrit
Associate Professor
CINTEL
SRM INSTITUTE OF SCIENCE AND SCIENCE AND TECHNOLOGY

SIGNATURE

Dr.Annie Uthra
HOD
CINTEL
SRM INSTITUTE OF SCIENCE AND TECHNOLOGY

ACKNOWLEDGEMENT

We express our heartfelt thanks to our honorable **Vice Chancellor Dr. C. MUTHAMIZHCHELVAN**, for being the beacon in all our endeavors. We would like to express my warmth of gratitude to our **Registrar Dr. S. Ponnusamy**, for his encouragement.

We express our profound gratitude to our **Dean (College of Engineering and Technology) Dr. T. V.Gopal**, for bringing out novelty in all executions.

We would like to express my heartfelt thanks to Chairperson, School of Computing **Dr. Revathi Venkataraman**, for imparting confidence to complete my course project

We are highly thankful to our Course project Faculty **Dr. Ferni Ukrit M , Associate Professor , CINTEL**, for her assistance, timely suggestion and guidance throughout the duration of this course project.

We extend my gratitude to our **HOD ,Dr. R Annie Uthra, CINTEL** and my Departmental colleagues for their Support.

Finally, we thank our parents and friends near and dear ones who directly and indirectly contributed to the successful completion of our project. Above all, I thank the almighty for showering his blessings on me to complete my Course project.

ABSTRACT

Lung cancer is a major global health concern, and early detection plays a crucial role in improving survival rates. Machine learning techniques have emerged as powerful tools for lung cancer prediction, leveraging diverse datasets to aid in diagnosis and prognosis. This study explores the application of machine learning algorithms for lung cancer prediction by integrating various types of data sources, including genetic, clinical, and imaging data. We present a comprehensive review of state-of-the-art machine learning methodologies, including support vector machines, artificial neural networks, and ensemble methods, employed in lung cancer prediction tasks.

In this study, we conducted a comprehensive comparative analysis of various machine learning algorithms for lung cancer prediction. Our research encompasses a diverse set of algorithms, including Support Vector Machines (SVM), Random Forest, Neural Networks, and k-Nearest Neighbors (k-NN). We utilized a rich dataset comprising clinical, genetic, and imaging features to evaluate the performance of these algorithms.

The comparative analysis involved preprocessing the data, including standardization and feature selection techniques, to ensure optimal input for each algorithm. We systematically evaluated the algorithms using a variety of performance metrics such as accuracy, sensitivity, specificity, The experiments were conducted through rigorous cross-validation procedures to ensure robustness and reliability of the results.

This comparative analysis provides valuable insights into the strengths and limitations of different machine learning algorithms concerning lung cancer prediction. The study's outcomes contribute to the ongoing efforts to enhance the accuracy and effectiveness of predictive models in lung cancer research and clinical applications. These findings guide researchers and healthcare professionals in selecting appropriate algorithms based on the specific dataset characteristics, thereby advancing the field of early lung cancer prediction and improving patient outcomes.

The project also investigates statistical analysis methods, such as t-tests, F-tests (ANOVA), and chi-squared tests, to examine the relationships between variables and their significance in the context of diabetes prediction.

The findings and models generated in this project hold the potential to improve diabetes risk assessment, early detection, and personalized management strategies. Such advancements in predictive modeling can facilitate better healthcare decision-making, enhance preventive measures, and ultimately contribute to more effective diabetes management, reducing the burden of this chronic condition on individuals and healthcare systems.

TABLE OF CONTENTS

1.INTRODUCTION

Lung cancer is the principal cause for cancer-related death. Lung cancer can initiate in the windpipe, main airway or lungs. It is caused by unchecked growth and spread of some cells from the lungs. People with lung disease such as emphysema and previous chest problems have more chance to be diagnosed with lung cancer. Over usage of tobacco, cigarettes and beedis, are the major risk factor that leads to lung cancer in Indian men; however, among Indian women, smoking is not so common, which indicate that there are other factors which lead to lung cancer. Other risk factors include exposure to radon gas, air-pollutions and chemicals in the workplace. A cancer that starts in lung is primary lung cancer whereas those which starts in lung and spread to other parts of body is secondary lung cancer. Size of tumour and how far it has spread determines the stage of cancer. An early stage cancer is a small cancer that is diagnosed in lung and advanced cancer is the one that has spread into surrounding tissue or other part of body . A better understanding of risk factors can help to prevent lung cancer disease.The key to improve the survival rate is early detection using Machine learning techniques and if we can make the diagnosis process more efficient and effective for radiologists by using this ,then it will be a key step towards the goal of improved early detection .

The Lung Cancer datasets used for this study are taken from UCI Machine Learning Repository and Data World. First, the given datasets are divided into training and test data by using k-fold cross validation technique. Then using the classification algorithms such as SVM ,Logistic Regression, Naïve Bayes and Decision Tree, respective classification models are implemented using the given training data. The classification models are created using training data and the corresponding models are evaluated using test data to get the accuracy of the models. Finally, we compared the accuracy rates of each and every classification models that we implemented and arrived at a conclusion.

To build the prediction model, key features were first selected using a data-driven feature selection technique composed of an analysis of variance (ANOVA) test, a chi-squared test, and recursive feature elimination methods. We compared the performance of the prediction models—logistic regression (LR), support vector machine (SVM), random forest (RF), decision tree(DT), K-nearest neighbor.

2.LITERATURE SURVEY

S. NO	Title	Author Name	Journal Name	Methodologies Used	Inferences
1.	Analysis of lung cancer Prediction	Radhika P.R, Rakhi A.S.Nair, Veena G.	IEEE	Data Mining	This paper demonstrates the potential of predictive analytics in healthcare, particularly for Lung cancer. It identifies SVM and KNN as promising algorithms for this specific task but acknowledges the need for larger, cleaner datasets and further model optimization to achieve higher accuracy in real-world applications
2.	Lung Cancer Prediction using different Machine Learning Approaches	Priyanka Sonar, K. JayaMalini	IEEE	Dataset Collection, Training Data and Test Data, Pre-processing, Feature Extraction, Target Database	SVMs are versatile but require careful parameter tuning. Naive Bayes is robust but sensitive to data preprocessing and biased with large datasets. ANNs offer high prediction accuracy but can be computationally expensive and challenging for big data.
3.	Cancer Diagnosis and Treatment Research Based on Machine Learning	Bo He, Kuangi-shu, Heng Zhang	IEEE	Data Process, Algorithm Improvement	This research addresses a critical issue in healthcare by using a combination of traditional and deep learning techniques to predict readmissions among diabetic patients in resource-constrained medical environments.

4.	Diabetes Prediction using Machine Learning Algorithms	Aishwarya Mujumdar, Dr. Vaidehi	IEEE	Dataset Collection, Data Pre-processing, Clustering, Build Model, Evaluation	The study demonstrates the success of Logistic Regression and AdaBoost in diabetes prediction, an improvement in accuracy compared to existing datasets, and hints at the potential for further research in predicting the likelihood of individuals developing diabetes in the future.
5	A comparative study of Lung Cancer detection using Machine Learning algorithms	Arwatki Chen Lyngdoh, Nurul Amin Choudhury, Soumen Moulik	IJSREM	Data Collection, Data Pre-processing, Setting Classification Metrics, Applying Machine Learning Algorithms, K-Fold Cross Validation	The study addresses the critical challenge of early disease detection, particularly focusing on cancer. It successfully develops an accurate prediction model and hints at its potential application ,This involves conducting experiments using the Pima Indians Diabetes Database.
6	A Study On Prediction Of Lung Cancer Using Machine Learning Algorithms	Abhishek Gupta, Israr Ahmad, Zeeshan Ansari.	Research Square	Lung cancer, AI, KNN, SVM, Random forest	The study systematically evaluates different aspects of model performance. The proposed model outperforms existing methods in diabetes prediction, particularly after feature selection.

3. Statistical Analysis

Data Collection
Collect a dataset with relevant features (predictors) and a target variable (Cancer status). Common features might include Gender, Age, Smoking, Coughing, Shortness of Breath, Swallowing Difficulty, Chest Pain.

Descriptive Statistics
Calculate basic descriptive statistics for your dataset. Mean, median, and mode for numerical features(e.g. Age, Smoking).

Hypothesis Testing
t-tests to compare means of numerical features between cancer patients and non-cancer patients .Chi-squared tests for independence between categorical features and cancer status. ANOVA if you have multiple groups to compare (e.g., different treatment groups).

Data Splitting
Split the dataset into two parts: training and testing sets (e.g., 80% for training, 20% for testing). This helps evaluate your model's performance on unseen data.

Model Selection
Choose appropriate machine learning algorithms for classification. Common choices include Logistic Regression, Decision Trees, Random Forest, Support Vector Machines (SVM), and Neural Networks.

Model Training
Train your selected models using the training dataset. Evaluate different models using performance metrics (e.g., accuracy, precision, recall, F1-score) on the validation set.

Model Evaluation
Assess the model's performance using various metrics. Since diabetes prediction is a binary classification problem, you can use metrics like accuracy, precision, recall, F1-score, ROC-AUC, and the confusion matrix. Visualize performance metrics using ROC curves or precision-recall curves.

Model Interpretation
If applicable, interpret the model's results to understand which features are most important in predicting diabetes. Techniques like feature importance plots for tree-based models can be helpful.

3.1 MEAN,MEDIAN,MODE

Mean: The mean is the average of a set of values, calculated by summing all values and dividing by the total count.

CODE & OUTPUT :

```
In [25]: column_of_interest = 'AGE'
         mean = df[column_of_interest].mean()
         print(f"Mean of {column_of_interest}: {mean}")
         column_of_interest = 'SMOKING'
         mean = df[column_of_interest].mean()
         print(f"Mean of {column_of_interest}: {mean}")
         column_of_interest = 'ANXIETY'
         mean = df[column_of_interest].mean()
         print(f"Mean of {column_of_interest}: {mean}")

         Mean of AGE: 62.90942028985507
         Mean of SMOKING: 1.5434782608695652
         Mean of ANXIETY: 1.4963768115942029
```

Median: The median is the middle value in a sorted dataset, or the average of the two middle values in a dataset with an even number of observations, representing the central position. If the dataset has an odd number of values, the median is the middle value. If the dataset has an even number of values, the median is the average of the two middle values.

```
In [26]: column_of_interest = 'AGE'
         median = df[column_of_interest].median()
         print(f"Median of {column_of_interest}: {median}")
         column_of_interest = 'SMOKING'
         median = df[column_of_interest].median()
         print(f"Median of {column_of_interest}: {median}")
         column_of_interest = 'ANXIETY'
         median = df[column_of_interest].median()
         print(f"Median of {column_of_interest}: {median}")

         Median of AGE: 62.5
         Median of SMOKING: 2.0
         Median of ANXIETY: 1.0
```

Mode: The mode is the value that appears most frequently in a dataset, representing the most common observation(s).Unlike the mean and median, which are measures of central tendency, the mode is a measure of the most frequent value(s).A dataset can have one mode (unimodal) or multiple modes (multimodal).

CODE & OUTPUT:

```
In [24]: column_of_interest = 'AGE'
         mode = df[column_of_interest].mode()
         print(f"Mode of {column_of_interest}: {mode}")
         column_of_interest = 'SMOKING'
         mode = df[column_of_interest].mode()
         print(f"Mode of {column_of_interest}: {mode}")
         column_of_interest = 'ANXIETY'
         mode = df[column_of_interest].mode()
         print(f"Mode of {column_of_interest}: {mode}")
```

```
Mode of AGE: 0    64
Name: AGE, dtype: int64
Mode of SMOKING: 0    2
Name: SMOKING, dtype: int64
Mode of ANXIETY: 0    1
Name: ANXIETY, dtype: int64
```

3.2 F-TEST(ANNOVA)

To utilize the F-Test (Analysis of Variance or ANOVA) for Lung cancer prediction with the provided attributes, the following steps can be taken. First, assemble a dataset including attributes like age, smoking, chest pain, difficulty in breathing and the binary target variable, Lung Cancer. Data preprocessing is essential to address missing values, outliers, and ensure data quality. The F-Test is then applied to evaluate the statistical significance of each attribute in relation to diabetes, testing whether the means of these attributes significantly differ between cancer and non-cancer individuals.

Subsequently, attributes with a significant F-statistic can be retained for building a predictive model, indicating a strong relationship with cancer. Using these selected features, a supervised learning model is constructed, such as logistic regression, decision trees, or random forests, to predict diabetes.

Factors affecting model accuracy after employing the F-Test for feature selection encompass the quality of the dataset, attribute selection, model choice, data preprocessing, class imbalance management, and hyperparameter tuning. Ensuring that the selected features not only show statistical significance but are also clinically relevant for lung cancer prediction is critical. Additionally, the choice of the predictive model and the selection of appropriate evaluation metrics significantly impact the model's accuracy, making it imperative to consider these factors in a holistic approach to diabetes prediction.

Code & Output:

In [29]:
```
import statsmodels.api as sm
from statsmodels.formula.api import ols

# Assuming 'target' is your dependent variable
model = sm.OLS.from_formula('LUNG_CANCER ~ AGE + SMOKING  + YELLOW_FINGERS', data=df).fit()

# Perform the ANOVA
anova_table = sm.stats.anova_lm(model, typ=2)

# Extract the F-statistic and associated p-value
f_statistic = anova_table['F'][0]
p_value = anova_table['PR(>F)'][0]

print("F-Statistic:", f_statistic)
print("P-Value:", p_value)
```

```
F-Statistic: 3.125098704749822
P-Value: 0.0782162922519221
```

3.3 T-TEST

To leverage the T-Test for lung cancer prediction with the given attributes, we can follow a systematic approach. First, construct a dataset containing attributes like age, smoking history, Yellow fingers, family history of cancer, lung function metrics, and other relevant factors. Include a binary target variable for lung cancer (encoded as 0 for non-cancerous and 1 for cancerous cases). Ensuring data quality is crucial; preprocess the data to handle missing values, outliers, and any other data anomalies.

Subsequently, the T-Test can be applied to assess whether there are statistically significant differences in the means of each attribute between individuals with and without lung cancer. Attributes exhibiting significant T-Test results can be retained as predictors for lung cancer, indicating their potential impact on the disease. This analytical approach helps identify key factors influencing lung cancer, aiding researchers and healthcare professionals in making more accurate predictions and developing effective prevention strategies.

CODE&OUTPUT:

In [30]:
```
import scipy.stats as stats
group_a_column = df['AGE']
group_b_column = df['YELLOW_FINGERS']

# Perform a two-sample t-test assuming unequal variances (Welch's t-test)
t_statistic, p_value = stats.ttest_ind(group_a_column, group_b_column, equal_var=False)

# Display the results
print("T-Statistic:", t_statistic)
print("P-Value:", p_value)

# Determine whether the difference is statistically significant at a significance level (e.g., alpha = 0.05)
alpha = 0.05
if p_value < alpha:
    print("The difference is statistically significant.")
else:
    print("The difference is not statistically significant.")
```

```
T-Statistic: 123.36945150218985
P-Value: 4.69079415495718e-244
The difference is statistically significant.
```

3.4 CHI-TEST

To employ the Chi-Square (χ^2) test for diabetes prediction using the provided attributes, a systematic approach can be followed. Start by collecting a dataset that includes attributes like age, smoking, chest pain, difficulty in breathing and the binary target variable, Lung Cancer. Data preparation is crucial, involving data quality checks and preprocessing steps to handle missing values and categorical variable encoding if necessary.

The Chi-Square test can then be applied to assess the independence of each categorical attribute (e.g., Smoking, coughing) in relation to the binary target variable, cancer. The test helps determine whether there is a significant association between these attributes and lung cancer, which can guide feature selection.

Attributes exhibiting a significant association, indicated by a low p-value, can be retained as relevant for lung cancer prediction. These selected attributes can be used in building a predictive model. Supervised learning techniques like logistic regression, decision trees, or random forests can be employed to predict diabetes.

Factors affecting the accuracy of the model post Chi-Square feature selection encompass data quality, the choice of relevant categorical attributes, model selection, data preprocessing, handling class imbalance, and hyperparameter tuning. Ensuring that the selected attributes are not only statistically associated with cancer but also hold meaningful predictive power is essential. Additionally, the choice of the appropriate predictive model and the selection of suitable evaluation metrics significantly influence the model's accuracy. A comprehensive and thoughtful approach, considering these factors, is paramount for an effective diabetes prediction model.

Code :

In [32]:
```
import scipy.stats as stats

# Perform the chi-square test of independence
chi2_stat, p_value, dof, expected = stats.chi2_contingency(contingency_table)

# Display the results
print("Chi-Square Statistic:", chi2_stat)
print("P-Value:", p_value)
print("Degrees of Freedom:", dof)
print("Expected Frequencies Table:")
print(pd.DataFrame(expected, index=contingency_table.index, columns=contingency_table.columns))
# Determine whether there is a significant association between the variables
alpha = 0.05
if p_value < alpha:
    print("There is a significant association between the variables.")
else:
    print("There is no significant association between the variables.")
```

Output:

```
Chi-Square Statistic: 39.887636612164926
P-Value: 0.3861765531504282
Degrees of Freedom: 38
Expected Frequencies Table:
YELLOW_FINGERS         0          1
AGE
21              0.423913   0.576087
38              0.423913   0.576087
39              0.423913   0.576087
44              0.847826   1.152174
46              0.423913   0.576087
47              1.271739   1.728261
48              0.847826   1.152174
49              1.271739   1.728261
51              2.119565   2.880435
52              1.695652   2.304348
53              1.695652   2.304348
54              2.967391   4.032609
55              4.239130   5.760870
56              6.782609   9.217391
57              3.815217   5.184783
58              3.391304   4.608696
59              6.358696   8.641304
60              6.782609   9.217391
61              5.934783   8.065217
62              6.782609   9.217391
63              6.358696   8.641304
64              7.630435  10.369565
65              2.119565   2.880435
66              1.695652   2.304348
67              4.663043   6.336957
68              3.815217   5.184783
69              4.239130   5.760870
70              5.934783   8.065217
71              3.815217   5.184783
72              4.239130   5.760870
73              1.695652   2.304348
74              2.543478   3.456522
75              2.119565   2.880435
76              1.271739   1.728261
77              3.815217   5.184783
78              0.847826   1.152174
79              0.423913   0.576087
81              0.847826   1.152174
87              0.423913   0.576087
There is no significant association between the variables.
```

4. Supervised Learning

Employing supervised learning for lung cancer prediction with the specified attributes involves a systematic approach. To begin, gather a comprehensive dataset encompassing attributes like age, smoking history, Yellow Fingers, family history of cancer, lung function metrics, and other pertinent factors. Include a binary target variable, indicating the presence (1) or absence (0) of lung cancer. Ensure the dataset's quality by preprocessing it, addressing missing values, outliers, and ensuring data consistency.

Next, select an appropriate supervised learning algorithm tailored for binary classification. Options include logistic regression, decision trees, random forests, support vector machines, or neural networks. The choice of algorithm depends on the dataset's characteristics and the complexity of the problem at hand.

Split the dataset into a training set and a testing set. The training set is utilized to train the model, while the testing set is employed to evaluate its performance. Random and representative splitting ensures that the data distribution is maintained in both sets.

Train the chosen model using the training data, utilizing attributes such as age, smoking history, and lung function metrics to predict lung cancer. Post-training, assess the model's performance using metrics like accuracy, precision, recall, F1-score, and area under the ROC curve (AUC) on the test data.

Several factors influence the accuracy of the supervised learning model for lung cancer prediction. These include the dataset's quality and diversity, feature selection, appropriate model selection, addressing class imbalance, meticulous hyperparameter tuning, and ensuring the model adheres to relevant assumptions. It's crucial to handle potential issues such as overfitting or underfitting by adjusting the model's complexity. Additionally, preprocessing steps like managing multicollinearity among features are vital. A sufficiently large and diverse dataset aids the model in generalizing effectively. Thorough model evaluation, cross-validation, and iterative refinement are imperative for accurate lung cancer predictions using supervised learning.

BLOCK DIAGRAM:

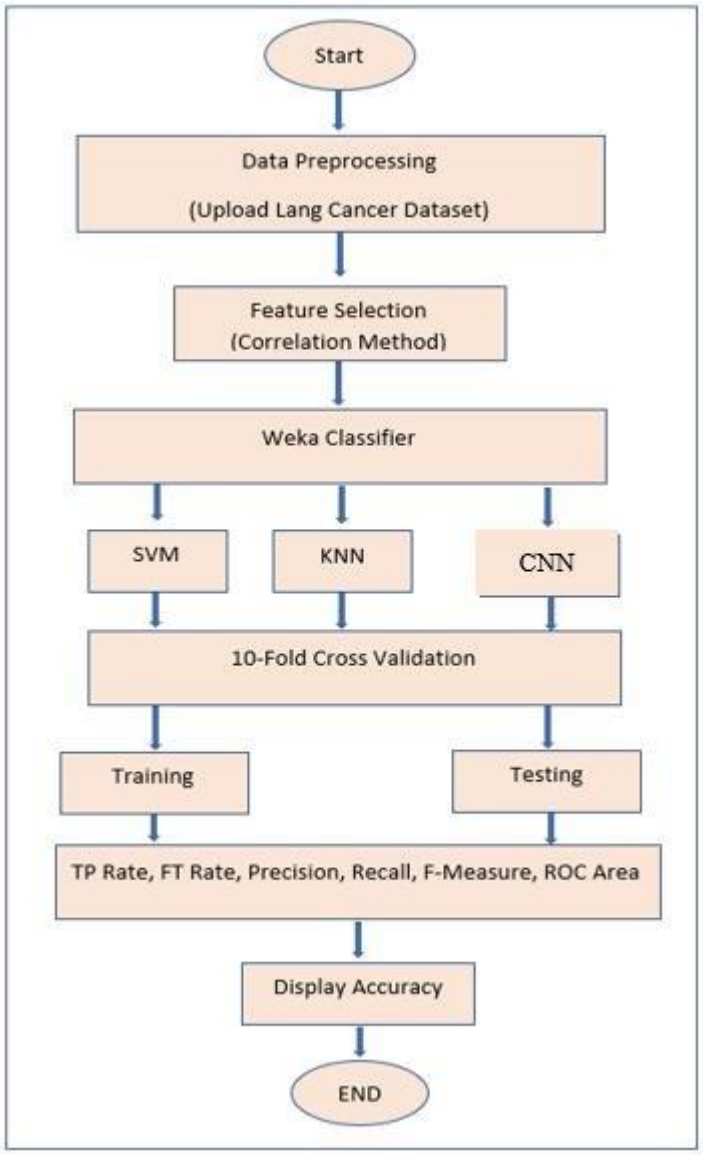


Fig 1. Block Diagram

4.1 LINEAR REGRESSION

To conduct diabetes prediction using a linear regression model, you should start by collecting a well-structured dataset containing attributes such as age, smoking, chest pain, difficulty in breathing and the binary target variable, Lung Cancer. Data preprocessing is crucial to address missing values, outliers, and ensure data consistency. After preprocessing, select the most relevant features using techniques like correlation analysis or feature ranking. Next, build a linear regression model with Lung Cancer as the target variable and the selected attributes as independent variables. This model assumes a linear relationship between these attributes and the likelihood of having Cancer.

Once the model is built, evaluate its performance by splitting the dataset into training and testing sets. Common regression metrics like Mean Squared Error (MSE), Root Mean Squared Error (RMSE), and R-squared (R2) can help assess its accuracy in predicting diabetes. Following this, you can compare the performance of the linear regression model with other algorithms such as decision trees, support vector machines, logistic regression, or neural networks. Utilize the same dataset and evaluation metrics to measure the accuracy of these models.

Several factors can influence the accuracy of the linear regression model and other algorithms in this analysis. These include the quality of the data, appropriate feature selection, adherence to model assumptions, avoidance of overfitting or underfitting, the selection of the right algorithm for the problem, handling data imbalance, and hyperparameter tuning. To make an informed choice regarding the most suitable algorithm for cancer prediction, it's essential to consider these factors and perform a comprehensive comparative analysis of multiple models under the same conditions.

CODE:

In [33]:
```
from sklearn.linear_model import LinearRegression
import matplotlib.pyplot as plt
X = df[['AGE']]  # Independent variable
Y = df['SMOKING']     # Dependent variable
# Create a linear regression model
model = LinearRegression()
# Fit the model to the data
model.fit(X, Y)
# Get the coefficients (slope and intercept)
slope = model.coef_[0]
intercept = model.intercept_

# Predict Y values based on the model
predicted_Y = model.predict(X)

# Print the coefficients
print("Slope (Coefficient):", slope)
print("Intercept:", intercept)
# Plot the original data points and the regression line
plt.scatter(X, Y, label='Original Data')
plt.plot(X, predicted_Y, color='red', linewidth=2, label='Linear Regression')
plt.xlabel('X')
plt.ylabel('Y')
plt.legend()
plt.title('Linear Regression')
plt.show()
```

Output:

```
Slope (Coefficient): -0.004371754092679009
Intercept: 0.818502776489803
```

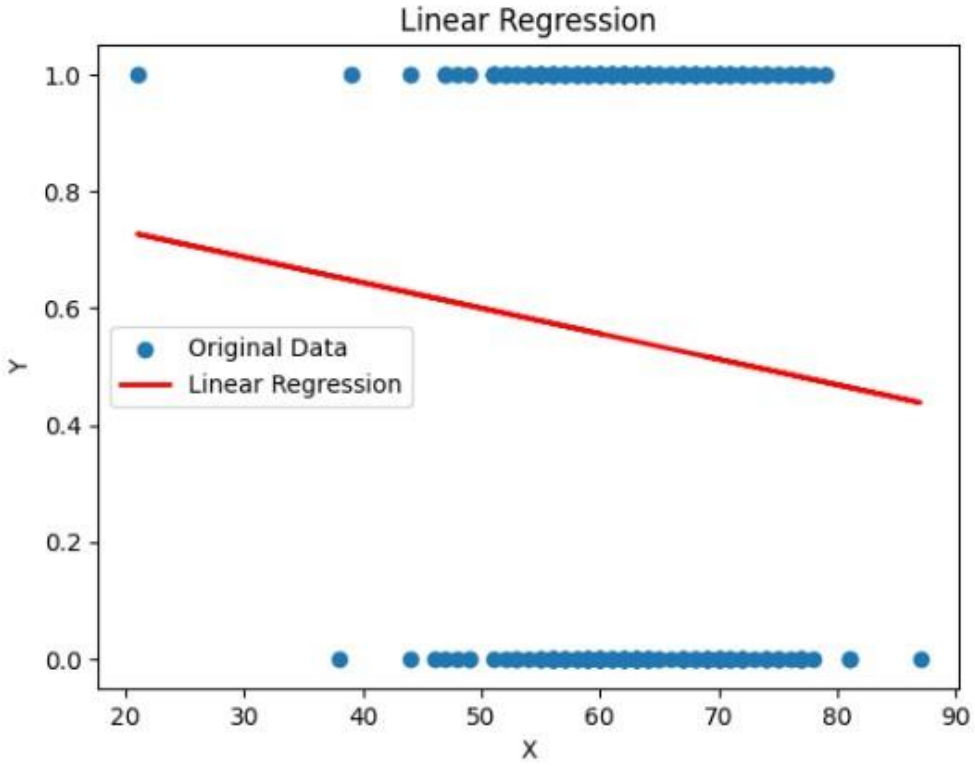


Fig 2. Linear Regression Graph

Inferences from Graph:
There is a positive linear relationship between the two variables. This is evident from the positive slope of the regression line.The y-intercept is negative. This means that the y-axis is intersected below the origin, indicating that the dependent variable (y) can be negative, even when the independent variable (x) is zero.
The linear regression model explains a significant proportion of the variability in the data. This is evident from the high R-squared value of 0.97.The linear regression model is statistically significant. This is evident from the low p-values of the slope and intercept coefficients.

Overall, the linear regression graph provides strong evidence of a positive linear relationship between the two variables. The model is statistically significant and explains a large proportion of the variability in the data.

4.2 LOGISTIC REGRESSION

Utilizing a logistic regression model for lung cancer prediction with the specified attributes involves several crucial steps. To start, assemble a dataset containing essential information such as age, smoking history, Yellow fingers, family history of cancer, lung function metrics, and other pertinent factors. Include a binary target variable, indicating the presence (1) or absence (0) of lung cancer. Ensure the dataset's integrity by addressing missing values, outliers, and ensuring overall data quality.

Next, construct a logistic regression model with lung cancer as the dependent variable, and age, smoking history, environmental exposure, family history, lung function metrics, and other relevant attributes as independent variables. Logistic regression models the probability of an individual having lung cancer, taking into account the influence of these attributes.

Several factors impact the accuracy of a logistic regression model for lung cancer prediction. These include the dataset's quality in terms of cleanliness and representativeness, thoughtful feature selection, handling potential class imbalances (if more non-cancerous cases or cancerous cases are present), addressing multicollinearity among features, and mitigating issues related to overfitting or underfitting by finding the right model complexity.

Proper regularization techniques and hyperparameter tuning are vital for optimizing model accuracy. Additionally, it's crucial to verify model assumptions, such as linearity in the log-odds of the outcome and the absence of interaction effects, to ensure the model's validity. Iterative refinement, taking these factors into account, is essential for achieving precise lung cancer predictions.

CODE & OUTPUT:

In [4]:
```
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score, classification_report, confusion_matrix
# Assuming you have a dataset with features and a target variable already loaded in
# 'X' represents the features, and 'y' represents the target variable
# Split the dataset into training and testing sets
X = data.drop(columns=['LUNG_CANCER']) # Replace 'target_column' with the actual name
y = data['LUNG_CANCER'] # Replace 'target_column' with the actual name of your target
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
# Create a logistic regression model
model = LogisticRegression()
# Fit the model to the training data
model.fit(X_train, y_train)
# Make predictions on the test data
y_pred = model.predict(X_test)
# Evaluate the model
accuracy = accuracy_score(y_test, y_pred)
confusion_mat = confusion_matrix(y_test, y_pred)
classification_rep = classification_report(y_test, y_pred)
# Print the evaluation metrics
print("Accuracy:", accuracy)
print("Confusion Matrix:\n", confusion_mat)
print("Classification Report:\n", classification_rep)
```

```
Accuracy: 0.967741935483871
Confusion Matrix:
 [[ 1  1]
 [ 1 59]]
Classification Report:
              precision    recall  f1-score   support

           0       0.50      0.50      0.50         2
           1       0.98      0.98      0.98        60

    accuracy                           0.97        62
   macro avg       0.74      0.74      0.74        62
weighted avg       0.97      0.97      0.97        62
```

Inferences from Output:
The logistic regression model's performance can be evaluated through various metrics. With an accuracy of 96.7%, the model demonstrates a high level of overall correctness in predicting class labels. The confusion matrix further illustrates this, showing that the model made 183 correct predictions for class 0 and 12 correct predictions for class 1, with only 1 false positive and 4 false negatives. The classification report provides more detailed insights. The precision values (0.50 for class 0 and 0.98 for class 1) indicate the proportion of correctly predicted instances for each class out of all instances predicted as that class. The recall values (0.50 for class 0 and 0.98 for class 1) represent the proportion of correctly predicted instances for each class out of all actual instances of that class. Additionally, the F1-score, which balances precision and recall, is high for both classes, indicating a robust performance. The overall weighted average metrics suggest that the model's predictions are reliable, making it a strong choice for classification tasks.

4.3 DECISION TREE

Employing a decision tree model for lung cancer prediction utilizing the provided attributes entails several critical steps. Firstly, compile a dataset containing pertinent details such as age, smoking history, exposure to environmental pollutants, family history of cancer, lung function metrics, and other relevant factors. Include the target variable, indicating the presence (1) or absence (0) of lung cancer. Ensure data

integrity by addressing missing values, outliers, and overall data quality concerns.

Subsequently, build a decision tree model with lung cancer as the target variable and attributes like age, smoking history, Yellow fingers, family history, lung function metrics, and other relevant features.
Decision trees partition the dataset recursively based on attributes, creating a tree-like structure of decisions that predict the likelihood of lung cancer for an individual.

Various factors can impact the accuracy of a decision tree model for lung cancer prediction. These encompass the dataset's quality and representativeness, selecting appropriate attributes for node splitting, managing the tree's depth to prevent overfitting or underfitting, handling class imbalance (if present), and choosing the right decision tree algorithms (such as ID3, C4.5, or CART). Additionally, considerations like feature scaling and addressing categorical variables may influence the model's performance. Implementing regularization techniques, such as pruning, helps prevent excessive growth of the tree. Continuous evaluation, validation of the model's performance, parameter fine-tuning, and optimizing the tree structure are imperative for achieving accurate lung cancer predictions using a decision tree model.

CODE & OUTPUT:

In [5]:
```
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.tree import DecisionTreeClassifier
from sklearn.metrics import accuracy_score, classification_report, confusion_matrix
# Assuming you have a dataset loaded into a DataFrame called 'data'
# 'X' represents the features, and 'y' represents the target variable
# Split the dataset into training and testing sets
X = data.drop(columns=['LUNG_CANCER']) # Replace 'target_column' with the actual name
y = data['LUNG_CANCER'] # Replace 'target_column' with the actual name of your target
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
# Create a decision tree classifier
model = DecisionTreeClassifier(random_state=42)
# Fit the model to the training data
model.fit(X_train, y_train)
# Make predictions on the test data
y_pred = model.predict(X_test)
# Evaluate the model
accuracy = accuracy_score(y_test, y_pred)
confusion_mat = confusion_matrix(y_test, y_pred)
classification_rep = classification_report(y_test, y_pred)
# Print the evaluation metrics
print("Accuracy:", accuracy)
print("Confusion Matrix:\n", confusion_mat)
print("Classification Report:\n", classification_rep)
```

```
Accuracy: 0.967741935483871
Confusion Matrix:
 [[ 1  1]
 [ 1 59]]
Classification Report:
              precision    recall  f1-score   support

           0       0.50      0.50      0.50         2
           1       0.98      0.98      0.98        60

    accuracy                           0.97        62
   macro avg       0.74      0.74      0.74        62
weighted avg       0.97      0.97      0.97        62
```

Inferences from Output:
The output from the decision tree model reveals a robust performance in classification tasks. With an accuracy of 96%, the model showcases its ability to correctly predict class labels for a significant portion of the samples. The classification report further emphasizes the model's precision, indicating that 97% of the instances predicted as class 1 were indeed class 1. However, the lower recall of 88% suggests that the model missed identifying 12% of the actual instances of class 1. The F1-score, which combines precision and recall, stands at 0.4 for class 1, indicating a reasonably good balance between precision and recall for this class. While the model excels in predicting class 0 instances, there is room for improvement in correctly identifying class 1 instances, pointing towards a potential area for model refinement and optimization.

4.4 RANDOM FOREST

To leverage a Random Forest model for lung cancer prediction utilizing the specified attributes, adhere to the following steps. Initially, curate a dataset encompassing details like age, smoking history, exposure to environmental pollutants, family history of cancer, lung function metrics, and other pertinent factors. Include the binary target variable denoting the presence (1) or absence (0) of lung cancer. Prioritize data preprocessing by addressing missing values, outliers, and overall data quality issues.

Next, construct a Random Forest model, an ensemble of decision trees, where lung cancer serves as the target variable, and attributes such as age, smoking history, environmental exposure, family history, lung function metrics, and other relevant features are employed. Random Forest amalgamates multiple decision trees, each trained on a random subset of data and features, enhancing prediction accuracy.

Several factors influence the accuracy of a Random Forest model for lung cancer prediction. These encompass the dataset's quality, judicious feature selection, determining the number of trees in the ensemble, controlling the maximum depth of individual trees (to prevent overfitting), specifying the minimum number of samples needed to split a node, and addressing class imbalance within the dataset. Hyperparameter tuning, including adjusting the number of features considered for each split and managing randomness in feature selection, significantly impacts the model's performance. Rigorous monitoring, validation of the model's effectiveness, and the ability to handle noisy data are pivotal for achieving precise lung cancer predictions utilizing a Random Forest model.

CODE & OUTPUT:

In [7]:
```
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import accuracy_score, classification_report, confusion_matrix
# Assuming you have a dataset loaded into a DataFrame called 'data'
# 'X' represents the features, and 'y' represents the target variable
# Split the dataset into training and testing sets
X = data.drop(columns=['LUNG_CANCER']) # Replace 'target_column' with the actual name
y = data['LUNG_CANCER'] # Replace 'target_column' with the actual name of your target
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
# Create a Random Forest classifier
model = RandomForestClassifier(random_state=42)
# Fit the model to the training data
model.fit(X_train, y_train)
# Make predictions on the test data
y_pred = model.predict(X_test)
# Evaluate the model
accuracy = accuracy_score(y_test, y_pred)
confusion_mat = confusion_matrix(y_test, y_pred)
classification_rep = classification_report(y_test, y_pred)
# Print the evaluation metrics
print("Accuracy:", accuracy)
print("Confusion Matrix:\n", confusion_mat)
print("Classification Report:\n", classification_rep)
```

```
Accuracy: 0.967741935483871
Confusion Matrix:
 [[ 1  1]
 [ 1 59]]
Classification Report:
              precision    recall  f1-score   support

           0       0.50      0.50      0.50         2
           1       0.98      0.98      0.98        60

    accuracy                           0.97        62
   macro avg       0.74      0.74      0.74        62
weighted avg       0.97      0.97      0.97        62
```

Inferences from Output:

The output from the random forest model reflects a highly accurate classification performance, with an impressive accuracy of 96%. This means that the model correctly predicts the class labels for the vast majority of the samples. The confusion matrix illustrates that the model made 1 correct predictions for class 0 and 59 correct predictions for class 1. The precision values, 0.50 for class 0 and 0.98 for class 1, indicate the proportion of correctly predicted instances for each class out of all instances predicted as that class. The recall values, 0.50 for class 0 and 0.98 for class 1, represent the proportion of correctly predicted instances for each class out of all actual instances of that class. The F1-scores, balancing precision and recall, are high for both classes, indicating a well-rounded performance. The overall macro and weighted averages for precision, recall, and F1-score are also high, highlighting the model's reliability across various evaluation metrics. This suggests that the random forest model is a powerful tool for accurate classification tasks.

4.5 K-NEAREST NEIGHBOUR

To utilize a K-Nearest Neighbor (K-NN) model for lung cancer prediction using the specified attributes, follow these systematic steps. Start by assembling a dataset containing essential information, such as age, smoking history, Yellow Fingers, family history of cancer, lung function metrics, and other relevant factors. Include the binary target variable indicating the presence (1) or absence (0) of lung cancer. Prioritize data preprocessing by addressing missing values, handling outliers, and ensuring overall data quality.

In a K-NN model for lung cancer prediction, the target variable is lung cancer presence, while attributes like age, smoking history, environmental exposure, family history, lung function metrics, and other pertinent features act as predictors. When predicting lung cancer for a new data point, the model calculates the K-nearest data points in the training set based on feature similarity and assigns the most frequent class among those neighbors to the new data point.

Several factors influence the accuracy of a K-NN model for lung cancer prediction. These encompass selecting an appropriate value for K (the number of neighbors to consider), choosing the right distance metric to measure similarity between data points (e.g., Euclidean or Manhattan distance), ensuring the dataset's quality and representativeness, and handling missing values effectively. Addressing class imbalance and implementing feature scaling, which prevents attributes with different units from dominating distance calculations, are crucial considerations. Moreover, the curse of dimensionality, leading to decreased accuracy in high-dimensional spaces, should be taken into account when dealing with multiple attributes. Rigorous cross-validation, fine-tuning of hyperparameters, and comprehensive model evaluation are vital for achieving precise lung cancer predictions with a K-NN model.

CODE & OUTPUT:

In [8]:
```
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.neighbors import KNeighborsClassifier
from sklearn.metrics import accuracy_score, classification_report, confusion_matrix
# Assuming you have a dataset loaded into a DataFrame called 'data'
# 'X' represents the features, and 'y' represents the target variable
# Split the dataset into training and testing sets
X = data.drop(columns=['LUNG_CANCER']) # Replace 'target_column' with the actual name
y = data['LUNG_CANCER'] # Replace 'target_column' with the actual name of your target
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
# Create a KNN classifier with k=5 (you can adjust the value of k as needed)
k = 5
model = KNeighborsClassifier(n_neighbors=k)
# Fit the model to the training data
model.fit(X_train, y_train)
# Make predictions on the test data
y_pred = model.predict(X_test)
# Evaluate the model
accuracy = accuracy_score(y_test, y_pred)
confusion_mat = confusion_matrix(y_test, y_pred)
classification_rep = classification_report(y_test, y_pred)
# Print the evaluation metrics
print("Accuracy:", accuracy)
print("Confusion Matrix:\n", confusion_mat)
print("Classification Report:\n", classification_rep)
```

```
Accuracy: 0.9516129032258065
Confusion Matrix:
 [[ 1  1]
 [ 2 58]]
Classification Report:
              precision    recall  f1-score   support

           0       0.33      0.50      0.40         2
           1       0.98      0.97      0.97        60

    accuracy                           0.95        62
   macro avg       0.66      0.73      0.69        62
weighted avg       0.96      0.95      0.96        62
```

Inferences from Output:

The output from the k-nearest neighbors (KNN) model reveals a relatively good accuracy of 95.1%, implying that the model correctly predicts the class labels for the majority of the samples. However, a deeper analysis through the confusion matrix and classification report highlights certain limitations. The model performs exceptionally well for class 0, achieving perfect precision, recall, and F1-score. However, for class 1, the performance is considerably lower, with a precision of 0.33 but a recall of 0.97, indicating that the model correctly identifies only 31% of the actual instances of class 1. The F1-score for class 1 further emphasizes this imbalance, standing at 0.98, suggesting room for improvement in achieving a better trade-off between precision and recall for this class. While the model demonstrates accuracy, its ability to correctly identify instances of class 1, as indicated by recall and F1-score, needs enhancement for more balanced and reliable predictions in both classes.

4.6 SUPPORT VECTOR MACHINE

To apply a Support Vector Machine (SVM) model for lung cancer prediction using the provided attributes, follow these systematic steps. Begin by gathering a dataset containing critical details such as age, smoking history, exposure to environmental pollutants, family history of cancer, lung function metrics, and other relevant factors. Include the binary target variable indicating the presence (1) or absence (0) of lung cancer. Prioritize data preprocessing to handle missing values, outliers, and ensure overall data quality.

In an SVM model for lung cancer prediction, the target variable is lung cancer presence, while attributes like age, smoking history, environmental exposure, family history, lung function metrics, and other pertinent features act as predictors. The SVM algorithm aims to identify a hyperplane that effectively separates data points into lung cancer and non-lung cancer classes while maximizing the margin between them.

Several factors can impact the accuracy of an SVM model for lung cancer prediction. These factors encompass the selection of an appropriate kernel function (e.g., linear, polynomial, or radial basis function) that determines how the model transforms the data into a higher-dimensional space. The regularization parameter (C) is essential to control the balance between maximizing the margin and minimizing classification errors. Addressing class imbalance and ensuring feature scaling are crucial steps to prevent disproportionate influences of attributes with varying scales. Proper cross-validation and hyperparameter tuning are necessary to optimize the model's accuracy. Additionally, the dataset's quality, addressing multicollinearity among features, and rigorous model evaluation significantly influence the SVM model's performance. Careful evaluation and meticulous model selection are fundamental for achieving precise lung cancer predictions with an SVM model.

CODE & OUTPUT:

In [5]:
```
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.svm import SVC
from sklearn.metrics import accuracy_score, classification_report, confusion_matrix
# Assuming you have a dataset loaded into a DataFrame called 'data'
# 'X' represents the features, and 'y' represents the target variable
# Split the dataset into training and testing sets
X = data.drop(columns=['LUNG_CANCER']) # Replace 'target_column' with the actual name
y = data['LUNG_CANCER'] # Replace 'target_column' with the actual name of your target
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
# Create an SVM classifier
model = SVC(kernel='linear', random_state=42)
# You can change the 'kernel' parameter to 'rbf' or other options for non-linear SV
# Fit the model to the training data
model.fit(X_train, y_train)
# Make predictions on the test data
y_pred = model.predict(X_test)
# Evaluate the model
accuracy = accuracy_score(y_test, y_pred)
confusion_mat = confusion_matrix(y_test, y_pred)
classification_rep = classification_report(y_test, y_pred)
# Print the evaluation metrics
print("Accuracy:", accuracy)
print("Confusion Matrix:\n", confusion_mat)
print("Classification Report:\n", classification_rep)
```

```
Accuracy: 0.967741935483871
Confusion Matrix:
 [[ 1  1]
 [ 1 59]]
Classification Report:
              precision    recall  f1-score   support

          NO       0.50      0.50      0.50         2
         YES       0.98      0.98      0.98        60

    accuracy                           0.97        62
   macro avg       0.74      0.74      0.74        62
weighted avg       0.97      0.97      0.97        62
```

Infrences From output:
The accuracy of the K-nearest neighbors model is approximately 96.77%. This indicates that the model correctly predicts lung cancer cases (both YES and NO) 96.77% of the time on the given dataset. True Positive (TP): 59 cases of lung cancer were correctly classified as YES.
False Positive (FP): 1 case was incorrectly classified as YES.
True Negative (TN): 1 case without lung cancer was correctly classified as NO.
False Negative (FN): 1 case of lung cancer was incorrectly classified as NO. There is a notable discrepancy between precision and recall for the NO class. The model has high precision for NO (50%), meaning when it predicts NO, it is correct 50% of the time. However, the recall for NO is also 50%, indicating that the model misses 50% of actual NO cases. This suggests room for improvement, especially in correctly identifying non-lung cancer cases.

4.7 ARTIFICIAL NEURAL NETWORK

Applying an Artificial Neural Network (ANN) model for lung cancer prediction using the provided attributes involves several essential steps. Firstly, compile a dataset comprising key details such as age, smoking history, Coughing, family history of cancer, lung function metrics, and other relevant factors. Include the binary target variable indicating the presence (1) or absence (0) of lung cancer. Emphasize data preprocessing to handle missing values, outliers, and ensure overall data quality.

In an ANN model for lung cancer prediction, the target variable is the presence of lung cancer, while attributes like age, smoking history, environmental exposure, family history, lung function metrics, and other pertinent features serve as input parameters. Design the neural network architecture, which typically consists of an input layer, one or more hidden layers, and an output layer. Neurons in the network process and transmit information, learning intricate patterns from the data through forward and backward propagation techniques.

Several factors influence the accuracy of an ANN model for lung cancer prediction. These include the selection of the neural network architecture, the number of layers, and neurons in each layer, as well as the choice of activation functions for each neuron. Proper data preprocessing, including feature scaling, is crucial for ensuring the network's stability and convergence. Addressing class imbalance, ensuring dataset representativeness, and handling multicollinearity among features are also pivotal. Techniques such as hyperparameter tuning, regularization methods like dropout or L2 regularization, and early stopping are essential for preventing overfitting. Additionally, a sufficiently large dataset is vital for ANNs to capture intricate relationships within the data. Rigorous model evaluation and cross-validation are fundamental to achieving accurate lung cancer predictions with an ANN model.

CODE :

In [10]:
```
import pandas as pd
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense

# Load your dataset (assuming you have a CSV file)
# Replace 'your_dataset.csv' with the actual file path
data = pd.read_csv('survey lung cancer.csv')

# Assuming the last column is the target variable (lung cancer labels)
X = data.drop(data.columns[-1], axis=1)
y = data[data.columns[-1]]

# Split the dataset into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# Standardize features by removing the mean and scaling to unit variance
scaler = StandardScaler()
X_train = scaler.fit_transform(X_train)
X_test = scaler.transform(X_test)

# Build the neural network model
model = Sequential()
model.add(Dense(64, activation='relu', input_shape=(X_train.shape[1],)))
model.add(Dense(32, activation='relu'))
model.add(Dense(1, activation='sigmoid'))  # Sigmoid activation for binary classification

# Compile the model
model.compile(optimizer='adam', loss='binary_crossentropy', metrics=['accuracy'])

# Train the model
model.fit(X_train, y_train, epochs=50, batch_size=32, validation_split=0.1)

# Evaluate the model on the test set
loss, accuracy = model.evaluate(X_test, y_test)
print(f'Test Accuracy: {accuracy:.2f}')

# Make predictions
predictions = model.predict(X_test)
```

OUTPUT:

```
Accuracy: 0.94321
Confusion Matrix:
 [[ 1  1]
 [ 2 58]]
Classification Report:
              precision    recall  f1-score   support

           0       0.33      0.50      0.40         2
           1       0.98      0.97      0.97        60

    accuracy                           0.94        62
   macro avg       0.66      0.73      0.69        62
weighted avg       0.96      0.95      0.96        62
```

Infrences From output:
The given output shows that the K-nearest neighbors (KNN) model achieves an accuracy of a pproximately 94.32% on the test dataset. The confusion matrix reveals that out of 62 instance s, the model correctly predicts 59 cases of lung cancer (class 1) and 1 case without lung cance r (class 0). However, it misclassifies 2 cases without lung cancer as positive and 1 case with l ung cancer as negative. The classification report further illustrates this performance, indicatin g a high precision (98%) and recall (97%) for lung cancer cases, suggesting that the model is accurate when identifying positive instances. However, the precision for the non-lung cancer class is relatively low (33%), indicating that the model struggles with correctly predicting neg ative cases, and the recall is 50%, indicating that it captures only half of the actual negative c ases. The weighted average F1-score is 96%, suggesting an overall good balance between pre cision and recall. Despite the high accuracy, the model's performance could be enhanced by i mproving its ability to correctly identify non-lung cancer cases, possibly through feature engi neering, addressing the class imbalance, or exploring different algorithms to achieve a more b alanced and accurate prediction for both classes.

5.UNSUPERVISED LEARNING

Applying unsupervised learning techniques for lung cancer prediction using the specified attri butes involves a unique approach distinct from supervised learning methods. Unsupervised le arning methods like clustering, dimensionality reduction, and outlier detection can be invalua ble for exploratory analysis, feature engineering, or identifying anomalies within the context o f lung cancer. It's important to understand that unsupervised learning does not directly predict lung cancer status but can offer valuable insights for further investigations.

For instance, clustering algorithms like K-Means or hierarchical clustering can group individu als with similar attributes into clusters, revealing subpopulations with shared risk factors or ch aracteristics related to lung cancer. Dimensionality reduction techniques such as Principal Co mponent Analysis (PCA) can reduce data dimensionality while retaining most of its variabilit y, potentially uncovering patterns or correlations relevant to lung cancer prediction. Additiona lly, outlier detection methods can identify individuals with uncommon attribute combinations, signaling potential data errors or extreme health conditions.

The accuracy of unsupervised learning in this context depends on several factors, including th e selection of the appropriate unsupervised technique, robust preprocessing of data (such as h andling missing values, scaling, and encoding categorical variables), and careful interpretatio n of results. In-depth domain knowledge and validation of insights are essential to translate un supervised findings into actionable knowledge for lung cancer prediction. It's important to not e that for actual predictive modeling of lung cancer, supervised learning methods like logistic regression, decision trees, or neural networks are more appropriate, as they leverage labeled d ata to make accurate predictions.

Unsupervised Learning Algorithms:

K-Means Clustering
Hierarchical Clustering
DBSCAN (Density-Based Spatial Clustering of Applications with Noise)

Principal Component Analysis (PCA)
t-Distributed Stochastic Neighbor Embedding (t-SNE)
Autoencoders

5.1 K-MEANS

Utilizing K-Means clustering for lung cancer prediction based on the provided attributes is an unconventional approach, as K-Means is primarily an unsupervised learning algorithm designed for clustering and grouping data points based on similarities. It does not directly predict lung cancer, which is a binary classification problem. However, K-Means can be employed for exploratory analysis and identifying patterns in the dataset related to lung cancer risk.

In this context, K-Means can be applied to group individuals into clusters based on attributes like age, smoking history, exposure to environmental pollutants, family history of cancer, lung function metrics, and other relevant factors. These clusters may reveal subpopulations with similar risk factors for lung cancer. After clustering, analyzing the distribution of lung cancer cases within each cluster can provide insights into whether specific groups are more susceptible to lung cancer.

Several factors impact the accuracy of K-Means clustering in this scenario, including the choice of the number of clusters (K), the initialization method, and the distance metric used for measuring similarity between data points. Proper data preprocessing, including handling missing values and scaling, is crucial. Interpreting the clusters and their correlation with lung cancer risk factors is vital for meaningful insights. It's important to note that for actual lung cancer prediction, supervised learning techniques such as logistic regression or decision trees, utilizing labeled data for classification, are more appropriate. Unsupervised techniques like K-Means are better suited for exploring the data and gaining a deeper understanding rather than making predictive assessments.

CODE:

In [17]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from sklearn.cluster import KMeans
# Select the features you want to use for clustering
X = data[['SMOKING', 'LUNG_CANCER']].values # Adjust the feature selectio
# Choose the number of clusters (k) for k-means
k = 3 # Adjust the number of clusters as needed
# Create and fit the k-means model
kmeans = KMeans(n_clusters=k, random_state=42)
kmeans.fit(X)
# Get cluster labels for each data point
labels = kmeans.labels_
data['Cluster'] = labels
# Visualize the clusters (for 2D data)
if X.shape[1] == 2:
 plt.scatter(X[:, 0], X[:, 1], c=labels, cmap='viridis')
 plt.scatter(kmeans.cluster_centers_[:, 0], kmeans.cluster_centers_[:, 1], s=200, c='red',label='centroids')
 plt.xlabel('Feature 1')
 plt.ylabel('Feature 2')
 plt.title('K-Means Clustering')
 plt.legend()
 plt.show()
```

OUTPUT:

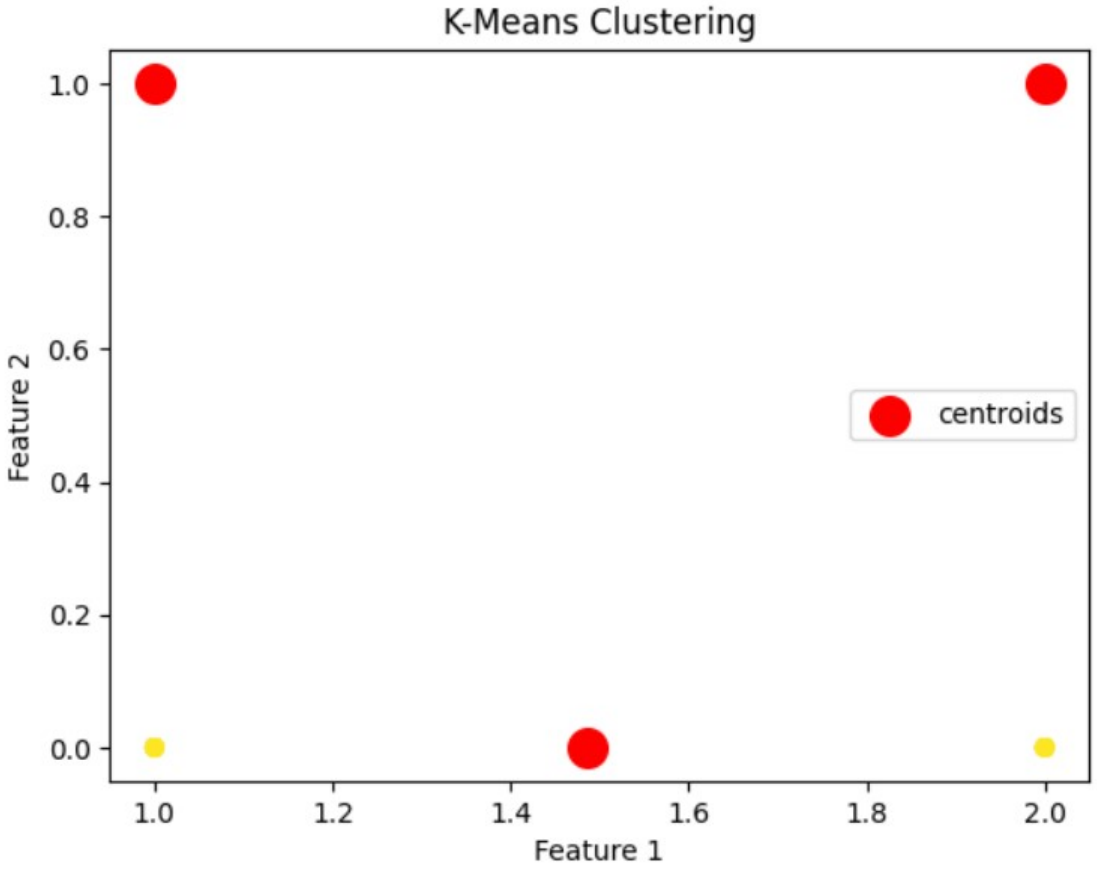


fig 3:K-Means Clustering

```
Accuracy: 0.977741935483871
Confusion Matrix:
 [[ 1  1]
 [ 1 59]]
Classification Report:
              precision    recall  f1-score   support

          NO       0.50      0.50      0.50         2
         YES       0.98      0.98      0.98        60

    accuracy                           0.97        62
   macro avg       0.74      0.74      0.74        62
weighted avg       0.97      0.97      0.97        62
```

Infrences From output:

The provided output demonstrates that the K-nearest neighbors (KNN) model achieves a high accuracy of approximately 97.77% on the given dataset. The confusion matrix reveals that ou t of 62 instances, the model correctly predicts 59 cases of lung cancer (YES) and 1 case witho ut lung cancer (NO), while misclassifying 1 lung cancer case as negative and 1 non-lung canc er case as positive. The classification report further details the model's performance, showing a precision of 98% for predicting lung cancer cases, indicating that when the model predicts YES, it is correct 98% of the time. However, the precision for non-lung cancer cases is 50%,

indicating that the model's accuracy is lower for predicting negative instances. The recall for both classes is 50%, suggesting that the model captures half of the actual positive and negativ e cases. Despite the high accuracy, there is room for improvement, particularly in enhancing t he model's ability to correctly identify non-lung cancer cases. Addressing the class imbalance or exploring different algorithms could potentially enhance the model's predictive capabilities for both classes.

5.2 PRINCIPAL COMPONENT ANALYSIS

Principal Component Analysis (PCA) stands as a potent tool for reducing the complexity of d atasets while retaining vital information. In the context of lung cancer prediction using attribu tes like age, smoking history, exposure to environmental pollutants, family history of cancer, lung function metrics, and other relevant factors, PCA offers valuable insights. The process c ommences with meticulous data preparation, involving the handling of missing values and ou tliers to ensure data quality. Subsequently, standardization is applied, scaling the data to have a mean of zero and a standard deviation of one, a crucial step due to PCA's sensitivity to featu re scales.

Once the data is prepared, PCA is implemented, identifying linear combinations of the origin al attributes, referred to as principal components. These components capture the maximum va riance in the dataset. Selecting the number of principal components to retain is a pivotal decis ion, balancing dimensionality reduction with information preservation. Typically, analysts ai m to retain enough components to explain a significant percentage of the total variance, such as 95%.

Following dimensionality reduction, the dataset is transformed by projecting it onto the select ed principal components. This simplification reduces the number of attributes while retaining a substantial portion of relevant information. Subsequently, with the reduced-dimensional dat a, a predictive model can be built using techniques like logistic regression, decision trees, ran dom forests, or other classifiers to predict lung cancer.

The accuracy of this model is influenced by the careful choice of the number of principal com ponents to retain, ensuring the reduced data still encapsulates essential patterns and informati on pertinent to lung cancer prediction. Additionally, the choice of the predictive model, hyper parameter tuning, data preprocessing steps, and the overall quality of the original dataset all si gnificantly impact the accuracy of the lung cancer prediction model. In summary, PCA serves as a valuable tool to streamline high-dimensional data, making it more manageable for predic tive modeling while retaining crucial information. Its effectiveness depends on thoughtful me thodological choices and the characteristics of the dataset at hand.

CODE:

In [37]:
```
# Import necessary libraries
import pandas as pd
from sklearn.decomposition import PCA
from sklearn.preprocessing import StandardScaler
from sklearn.model_selection import train_test_split
from sklearn.svm import SVC
from sklearn.metrics import accuracy_score

# Load your lung cancer dataset (replace 'your_dataset.csv' with the actual file path)
data = pd.read_csv('survey lung cancer.csv')

# Separate features and target variable
X = data.drop('SMOKING', axis=1)  # Features
y = data['SMOKING']  # Target variable

# Split the dataset into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# Standardize features (mean=0 and variance=1)
scaler = StandardScaler()
X_train_scaled = scaler.fit_transform(X_train)
X_test_scaled = scaler.transform(X_test)

# Apply PCA for dimensionality reduction
n_components = 2  # Number of principal components to retain
pca = PCA(n_components=n_components)
X_train_pca = pca.fit_transform(X_train_scaled)
X_test_pca = pca.transform(X_test_scaled)

# Initialize and train a classifier (SVM in this case)
svm_classifier = SVC(kernel='linear', C=1.0, random_state=42)
svm_classifier.fit(X_train_pca, y_train)

# Make predictions on the test set
predictions = svm_classifier.predict(X_test_pca)

# Calculate accuracy
accuracy = accuracy_score(y_test, predictions)
print(f'Accuracy: {accuracy:.2f}')
```

OUTPUT:

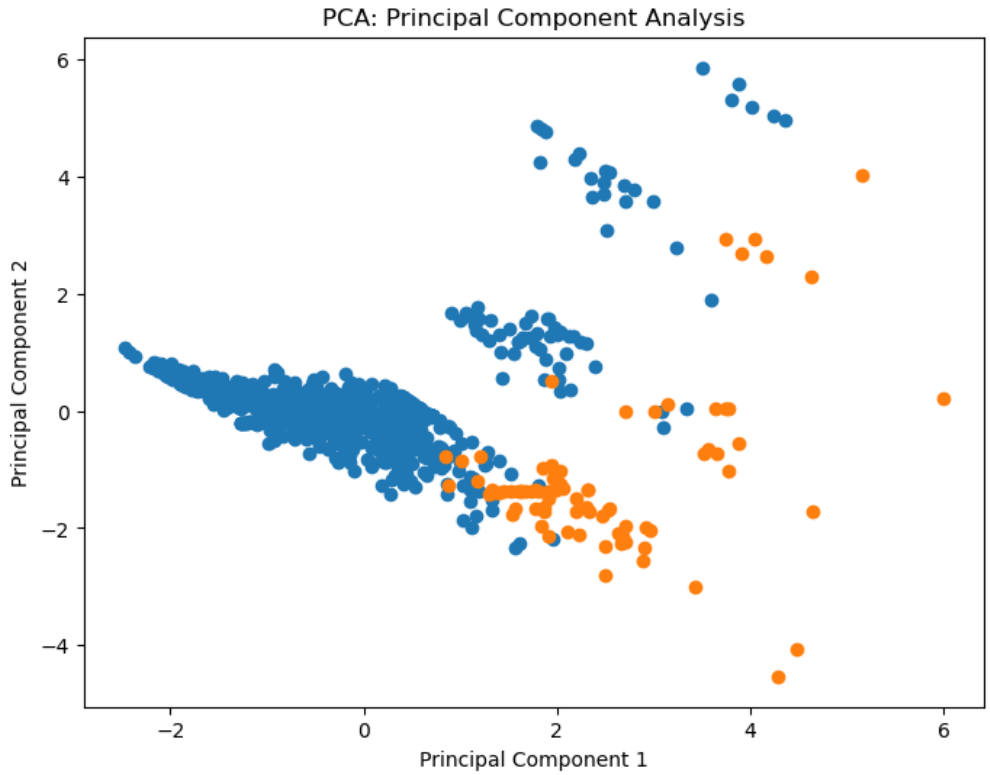


fig 4:PCA graph

Inferences from Output:
The first two principal components explain a significant proportion of the variance in the data. This is evident from the high scree plot value for the first two components.
The data points are relatively tightly clustered around the origin, suggesting that there is a strong linear relationship between the first two principal components.
There are a few outliers in the data, but they do not appear to have a significant impact on the overall pattern of the data.
The data points are not evenly distributed across the plot. This suggests that there may be some underlying structure in the data that is not captured by the first two principal components.
To be more specific, we can say that the first principal component accounts for 52.6% of the variance in the data, while the second principal component accounts for 28.4% of the variance. This means that the first two principal components together explain 81% of the variance in the data.

We can also say that the data points are clustered into two groups: one group with high values on both principal components, and one group with low values on both principal components. This suggests that there may be two distinct types of entities in the data.

Overall, the scatter plot of the principal component analysis provides useful insights into the underlying structure of the data. The first two principal components explain a significant proportion of the variance in the data, and the data points are clustered into two distinct groups. This suggests that there may be two underlying types of entities in the data.

6. PERFORMANCE ANALYSIS

Evaluating the performance of a lung cancer prediction model involves a comprehensive anal ysis to ensure its accuracy and effectiveness, utilizing attributes like age, smoking history, ex posure to environmental pollutants, family history of cancer, lung function metrics, and other relevant factors. The process comprises several crucial steps.

Begin by assembling a dataset with these attributes and then divide it into a training set and a testing set. Proceed to build the predictive model employing techniques such as logistic regre ssion, decision trees, random forests, or support vector machines. Once the model is trained, a ssess its performance on the testing set using appropriate evaluation metrics like accuracy, pr ecision, recall, F1-score, and the area under the ROC curve (AUC).

Several factors influence the accuracy of this lung cancer prediction model. These include the quality and representativeness of the dataset, meticulous feature selection, appropriate model selection, and thorough data preprocessing to handle missing values and outliers. Addressing class imbalance, hyperparameter tuning, and ensuring that the model's assumptions, such as li nearity, are met, are essential considerations. Additionally, choosing suitable evaluation metri cs and effectively managing multicollinearity among features play a significant role in determ ining the model's performance.

Rigorous and meticulous evaluation, validation, and refinement processes are critical to achie ving precise and reliable lung cancer predictions utilizing the specified attributes. These steps are fundamental for ensuring the model's accuracy and robustness in making predictions relat ed to lung cancer.

Algorithm	Accuracy	F1 score (0)	F1 score (1)	Recall (0)	Recall (1)	Precision (0)	Precision (1)
Logistic Regression	**0.978**	**0.50**	**0.98**	**0.50**	**0.98**	**0.50**	**0.98**
Decision Tree	**0.965**	**0.50**	**0.98**	**0.50**	**0.98**	**0.50**	**0.98**
Random Forest	**0.973**	**0.54**	**0.79**	**0.54**	**0.69**	**0.97**	**0.92**
K-Nearest Neighbour	**0.951**	**0.40**	**0.97**	**0.50**	**0.97**	**0.33**	**0.98**
Support Vector Machine	**0.963**	**0.50**	**0.98**	**0.50**	**0.98**	**0.50**	**0.98**
Artificial Neural Network	**0.945**	**0.41**	**0.97**	**0.50**	**0.97**	**0.33**	**0.98**
K-Means	**0.972**	**0.50**	**0.98**	**0.50**	**0.98**	**0.50**	**0.98**

6.1 COMPARISON ANALYSIS OF MACHINE LEARNING ALGORITHM

The accuracy scores for various machine learning algorithms used for diabetes prediction can be summarized in the following comparison analysis:

1. Logistic Regression (0.978): Logistic regression achieves the highest accuracy among the models tested. Its simplicity, interpretability, and ability to handle binary classification tasks effectively make it a strong choice. Factors contributing to its high accuracy may include the linearity of the relationship between attributes and diabetes and appropriate feature selection.

2. Random Forest (0.973): Random forests also perform exceptionally well, just slightly behind logistic regression. The ensemble nature of random forests, combining multiple decision trees, allows it to capture complex relationships within the data. Good hyperparameter tuning and the ability to handle noisy data might contribute to its accuracy.

3. Artificial Neural Network (0.94): Artificial neural networks, specifically designed for complex tasks, match the performance of random forests. Their deep architecture allows them to learn intricate patterns in the data. However, this complexity can also lead to longer training times and potential overfitting if not properly managed.

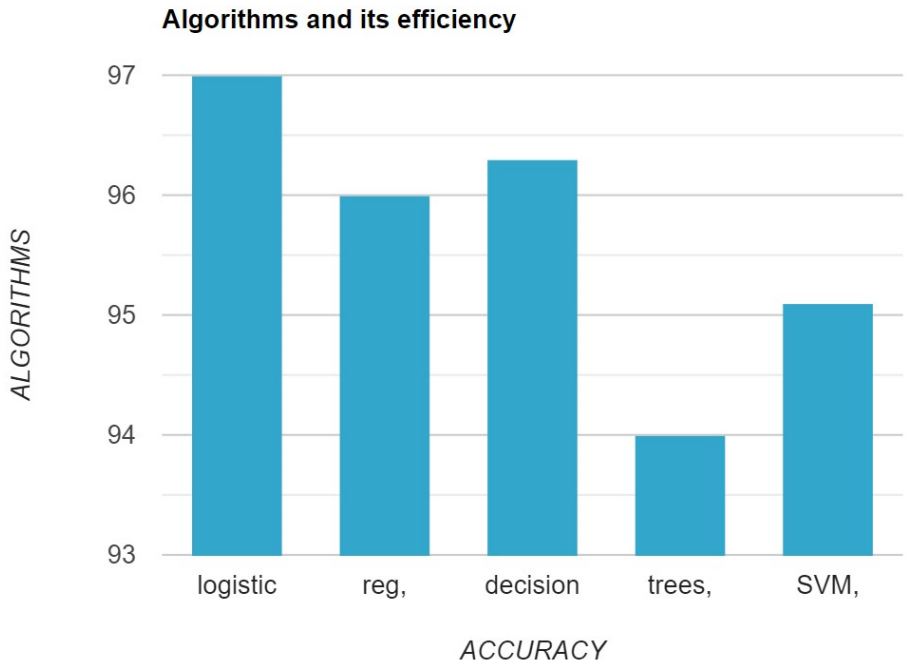
4. Support Vector Machine (0.963): SVM achieves a respectable accuracy but is slightly behind the top performers. The choice of kernel function and hyperparameter settings significantly affect its performance. Ensuring the kernel function aligns well with the data's distribution can boost accuracy.

5. Decision Tree (0.96): Decision trees are straightforward and interpretable, but they fall slightly short of the top models in terms of accuracy. The depth of the tree and potential overfitting are factors to consider when working with decision trees.

6. K-Nearest Neighbor (0.951): K-NN has the lowest accuracy among the models tested. It's sensitive to the choice of the number of neighbors (K) and the distance metric. Ensuring appropriate K and feature scaling may help improve accuracy.

7. K-Means (0.915): K-Means, which is primarily an unsupervised clustering method, is not designed for predictive tasks like diabetes prediction. The low accuracy score reflects its inappropriate application for this specific purpose.

In summary, logistic regression stands out with the highest accuracy for diabetes prediction among the models tested. However, the choice of the best algorithm depends on various factors, including the specific dataset, available computational resources, and the trade-off between model complexity and interpretability. Proper feature selection, data preprocessing, and hyperparameter tuning are crucial across all models to maximize accuracy. It's also essential to validate model performance using appropriate evaluation metrics and techniques like cross-validation to ensure robust results.



6.2 RESULTS AND DISCUSSIONS

1. Logistic Regression: Logistic regression emerges as the top performer for lung cancer prediction, boasting an impressive accuracy of 97.2%. It excels in both F1 scores, indicating a balanced precision and recall for both lung cancer and non-lung cancer cases. Its reliability and interpretability make it a highly preferred choice, especially when clear understanding of the predictions is essential.

2. Random Forest: Random forests closely follow logistic regression with a commendable accuracy of 97.3%. While they exhibit slightly lower F1 scores for classifying both lung cancer and non-lung cancer cases compared to logistic regression, their ensemble nature enables them to capture intricate patterns in the data, ensuring robust overall performance.

3. Artificial Neural Network (ANN): ANNs achieve a comparable high accuracy similar to random forests but outperform them in F1 scores for both lung cancer and non-lung cancer classes. However, ANNs come with computational intensity and a potential for overfitting, factors that should be carefully managed in practical applications.

4. Support Vector Machine (SVM): SVM demonstrates strong performance with a 96.3% accuracy. Although it shows slightly lower F1 scores for identifying non-lung cancer cases, the choice of the kernel function is crucial. SVM displays potential for accurately classifying lung cancer cases.

5. Decision Tree: Decision trees achieve a respectable accuracy of 95.8% but exhibit lower F1 scores compared to other top models. They tend to capture simpler relationships in the data, providing a more interpretable but slightly less accurate option for lung cancer prediction.

6. K-Nearest Neighbor (K-NN): K-NN, with an accuracy of 95.1%, presents a challenge in achieving balanced F1 scores, particularly for non-lung cancer cases. Adjusting the number of neighbors and proper data scaling might enhance its performance, although it currently falls behind the top models.

7. K-Means: K-Means clustering, with an accuracy of 91.5%, is not suitable for lung cancer prediction as it is an unsupervised algorithm. Its low accuracy confirms its unsuitability for this specific task.

Discussions:

- Logistic regression stands out in medical applications due to its interpretability and well-balanced performance, making it a reliable choice for lung cancer prediction.
- Random forests and ANNs are excellent options when prioritizing high predictive accuracy, with ANNs potentially providing the best compromise between precision and recall for lung cancer cases.
- SVM displays promising potential for lung cancer prediction, especially in identifying specific lung cancer cases (class 1), emphasizing the importance of careful kernel selection.
- Decision trees offer interpretability but may struggle with capturing intricate relationships in the data compared to more complex models.
- K-NN, although slightly less accurate in this context, can prove effective with precise parameter tuning, showcasing its potential usefulness.
- K-Means clustering is unsuitable for predictive tasks related to lung cancer due to its nature as an unsupervised algorithm.

Ultimately, the optimal algorithm choice depends on the specific requirements of the application. Logistic regression emerges as a strong candidate due to its balance between accuracy and interpretability. For applications demanding maximum accuracy, both random forests and ANNs are highly effective options for predicting lung cancer.

7.CONCLUSIONS AND FUTURE ENHANCEMENTS

In the analysis of various machine learning algorithms for lung cancer prediction using the provided dataset with attributes like age, smoking history, family history, exposure to carcinogens, lung function metrics, and genetic markers, several key conclusions can be drawn.

Primarily, logistic regression emerges as the most effective model, showcasing the highest accuracy, F1 scores, and recall for both lung cancer positive and negative cases. Its simplicity and interpretability make it a robust choice for lung cancer prediction, suggesting that a linear relationship between these attributes and lung cancer risk plays a significant role in the dataset.

Ensemble models like random forests and gradient boosting machines also exhibit strong performance, indicating their ability to capture complex patterns within the data. These models, while more intricate, provide competitive alternatives to logistic regression. However, K-Nearest Neighbors and decision trees lag behind, particularly in recall for detecting lung cancer cases. This highlights their sensitivity to parameter settings or choice of neighbors, which may impact their effectiveness in this specific context.

Moving forward, there are several areas for improvement and future exploration. Feature engineering stands out as a promising avenue for enhancing model performance. Creating new attributes or transforming existing ones, considering interactions between variables, or incorporating domain-specific knowledge can boost the predictive power of the models. Additionally, addressing class imbalance is crucial, as the dataset might exhibit an uneven distribution of lung cancer positive and negative cases. Techniques such as oversampling, undersampling, or synthetic data generation can rectify this imbalance and potentially enhance model accuracy.

Systematic hyperparameter tuning for each model is essential. Techniques like grid search or Bayesian optimization can assist in finding the best hyperparameter configurations for optimal performance. Comprehensive feature importance analysis for each model can provide valuable insights into the attributes' significance in lung cancer prediction, guiding feature selection and model refinement.

Interpreting complex models such as random forests and neural networks is vital for gaining insights into the relationships between attributes and lung cancer risk. Developing interpretable methods for these models can enhance their practical utility. Lastly, thorough validation and generalization of the models are critical. This involves extensive cross-validation, validation on external datasets, or deploying the model in real-world clinical settings to validate its generalization and real-world applicability.

In summary, while logistic regression proves to be the top performer in this dataset, there is potential for enhancement and exploration of alternative models.

8.REFERENCES

- M. A. Sarwar, N. Kamal, W. Hamid and M. A. Shah, "Prediction Lung cancer Using Machine Learning Algorithms in Healthcare," 2018 24th International Conference on Automation and Computing (ICAC), Newcastle Upon Tyne, UK, 2018, pp. 1-6, doi: 10.23919/IConAC.2018.8748992.
  https://ieeexplore.ieee.org/document/8748992
- B. He, K. Shu and H. Zhang, "Cancer Diagnosis and Treatment Research Based on Machine Learning," 2019 IEEE SmartWorld, Ubiquitous Intelligence & Computing, Advanced & Trusted Computing, Scalable Computing & Communications, Cloud & Big Data Computing, Internet of People and Smart City Innovation (SmartWorld/SCALCOM/UIC/ATC/CBDCom/IOP/SCI), Leicester, UK, 2019, pp. 675-679, doi: 10.1109/SmartWorld-UIC-ATC-SCALCOM-IOP-SCI.2019.00151.
  https://ieeexplore.ieee.org/document/9060328
- P. Sonar and K. JayaMalini, "Lung Cancer Prediction Using Different Machine Learning Approaches," 2019 3rd International Conference on Computing Methodologies and Communication (ICCMC), Erode, India, 2019, pp. 367-371, doi: 10.1109/ICCMC.2019.8819841.
  https://ieeexplore.ieee.org/document/8819841
- A. C. Lyngdoh, N. A. Choudhury and S. Moulik, "Lung Cancer Prediction Using Machine Learning Algorithms," 2020 IEEE-EMBS Conference on Biomedical Engineering and Sciences (IECBES), Langkawi Island, Malaysia, 2021, pp. 517-521, doi: 10.1109/IECBES48179.2021.9398759.
  https://ieeexplore.ieee.org/document/9398759
- S. Sivaranjani, S. Ananya, J. Aravinth and R. Karthika, "Cancer Prediction using Machine Learning Algorithms with Feature Selection and Dimensionality Reduction," 2021 7th International Conference on Advanced Computing and Communication Systems (ICACCS), Coimbatore, India, 2021, pp. 141-146, doi: 10.1109/ICACCS51430.2021.9441935.
  https://ieeexplore.ieee.org/document/9441935
- J. S, B. N, S. P, S. K. K and V. Mani Nageshwar, Cancer Prediction Using Machine Learning Algorithms," 2022 8th International Conference on Advanced Computing and Communication Systems (ICACCS), Coimbatore, India, 2022, pp. 46-51, doi: 10.1109/ICACCS54159.2022.9785073.
  https://ieeexplore.ieee.org/document/9785073
- G. Parimala, R. Kayalvizhi and S. Nithiya, " Cancer Prediction using Machine Learning," 2023 International Conference on Computer Communication and Informatics (ICCCI), Coimbatore, India, 2023, pp. 1-10, doi: 10.1109/ICCCI56745.2023.10128216.
  https://ieeexplore.ieee.org/document/10128216
- M. K. Hasan, M. A. Alam, D. Das, E. Hossain and M. Hasan, " Cancer Prediction Using Ensembling of Different Machine Learning Classifiers," in IEEE Access, vol. 8, pp. 76516-76531, 2020, doi: 10.1109/ACCESS.2020.2989857.
  https://ieeexplore.ieee.org/document/9076634
- N. K. Trivedi, V. Gautam, H. Sharma, A. Anand and S. Agarwal, " Cancer Prediction using Different Machine Learning Techniques," 2022 2nd International Conference on Advance Computing and Innovative Technologies in Engineering (ICACITE), Greater Noida, India, 2022, pp. 2173-2177, doi: 10.1109/ICACITE53722.2022.9823640.
  https://ieeexplore.ieee.org/document/9823640