INTERNSHIP PROJECT REPORT

ON

LUNG CANCER PREDICTIONS

SUBMITTED BY,

MEGHASHREE K

(4VP21AI030)

AT

Zephyr Technologies and Solutions Pvt Ltd.



5 th FLOOR, OBERLE TOWERS, BALMATTA RD, BENDOOR,

MANGALORE, KARNATAKA 575002, INDIA.

**ACKNOWLEDGEMENT**

We dedicate this page to acknowledge and thank those responsible for the shaping of the project. Without their guidance and help, the

experience while constructing the dissertation would not have been so smooth and efficient.

We would like to take the opportunity to express our humble gratitude to employees of Zephyr technologies under whom we executed this

project.

Their constant guidance and willingness to share their vast knowledge made us understand this project and its manifestations in great depth and helped us to complete the assigned tasks.

Finally, yet importantly, we would also like to extend thanks to our beloved parents for their blessings, friends and classmates for their help and wishes for the successful completion of the project.

………….

**ABSTRACT**

Lung cancer is the most common form of cancer in humans and a leading cause of cancer-related deaths worldwide. Early detection and prognosis are critical for improving survival rates. This research aims to evaluate and compare two machine learning algorithms, particularly Logistic Regression (LR), for predicting lung cancer survival outcomes. The primary objective is to assess the accuracy, confusion matrix, data representation in forecasting patient survival based on clinical data.

The study utilizes Python and essential machine learning libraries such as Pandas, NumPy, and Scikit-learn to analyze the dataset and uncover meaningful patterns. Logistic regression is employed as a predictive model due to its effectiveness in binary classification problems like cancer prediction. Performance evaluation metrics such as accuracy, confusion matrix, and error scores (mean squared error, R2 score) are used to measure the model's effectiveness. These metrics are critical in understanding how well the model performs in predicting lung cancer outcomes, providing valuable insights for healthcare professionals.

The project also emphasizes the significance of a user-friendly interface that allows clinicians or users to input patient parameters and receive predictions regarding lung cancer likelihood. This functionality enhances the model's practical application in real-world settings, assisting healthcare providers in making accurate prognoses.

In conclusion, this study highlights the potential of logistic regression as a tool for lung cancer survival prediction, while suggesting that future iterations could incorporate more advanced algorithms and larger datasets to further refine the model's accuracy and utility. The findings contribute to ongoing efforts to improve lung cancer diagnosis and patient care through machine learning.

………….

**INDEX**

1. **INTRODUCTION………………………………….05**
2. **BACKGROUND STUDY………………………….06**
3. **PROBLEM STATEMENT………………………...07**
4. **DATASET…………………………………………..08**
5. **METHODOLOGY…………………………………10**
6. **REQUIREMENTALGORITHMS………………..12**
7. **IMPLEMENTATION……………………………...14**
8. **RESULT AND DISCUSSION……………………..18**
9. **CONCLUSION……………………………………..21**

**10. SCOPE FOR FUTURE ENHANCEMENT……...22**

**11. APPENDIX…………………………………………25**

**12. BIBILOGRAPHY………………………………….28**

**INTRODUCTION**

Lung cancer is one of the most lethal respiratory diseases, with increasing global mortality rates largely linked to air pollution and smoking. While advancements in medicine have improved detection and treatment options, early diagnosis of lung cancer remains a significant challenge due to the subtle nature of initial symptoms. In the early stages, symptoms such as persistent cough, shortness of breath, back pain, and unexplained weight loss may be present, but they are often overlooked or attributed to less serious conditions. Early diagnosis is crucial for effective treatment and improved patient outcomes, yet detecting lung cancer at this stage is difficult due to the complex interaction of various risk factors, including environmental exposures and lifestyle habits.

Recent advancements in health data storage and analysis have paved the way for using machine learning techniques to assist in the early detection of lung cancer. By applying these computational methods to large, complex datasets, researchers and clinicians can identify patterns that may go unnoticed through traditional diagnostic approaches. However, the accuracy of these predictions relies heavily on the availability and quality of data, as well as the careful selection of key features for analysis.

This study aims to leverage machine learning techniques to improve the early prediction of lung cancer based on patient symptoms, habits, and environmental factors. Specifically, the research utilizes Seaborn for data visualization, along with Linear Regression and Logistic Regression for predictive modelling. By comparing the performance of these algorithms, we aim to assess their ability to accurately predict lung cancer outcomes and assist clinicians in diagnosing the disease earlier, when treatment is most effective.

The analysis explores various machine learning models, focusing on the accuracy, precision, recall, and F1 score of the predictions. The results will help determine which algorithm performs best in identifying lung cancer at an early stage, potentially leading to better treatment options and improved survival rates. This research not only underscores the importance of early detection but also demonstrates how machine learning can enhance lung cancer prognosis by integrating clinical and environmental data into predictive models.

**BACKGROUND STUDY**

Traditionally, doctors relied on medical imaging techniques such as X-rays, CT scans, and biopsies, combined with clinical evaluations, to detect and diagnose lung cancer. These methods often required significant expertise and were typically used when the disease had already advanced, limiting early detection and treatment opportunities. Physicians also relied on patients' medical histories, physical examinations, and symptom observation. Despite their importance, these traditional approaches often resulted in late-stage diagnosis due to the subtle nature of early symptoms, reducing the chances of successful treatment.

With advancements in data storage and analysis, machine learning (ML) techniques are transforming lung cancer diagnosis. ML algorithms analyze large volumes of clinical and imaging data, identifying patterns that may be missed by human interpretation. These methods can predict the likelihood of cancer by examining various factors such as patient demographics, medical history, and symptoms. Logistic regression, decision trees, and neural networks are among the most commonly used ML models for cancer prediction. These models excel in processing large datasets and handling complex relationships between variables.

In particular, logistic regression has proven effective in binary classification tasks like lung cancer prediction. By preprocessing the data and applying performance metrics like accuracy, precision, and recall, ML models can provide more accurate early-stage predictions. This shift from traditional diagnostic methods to machine learning enhances early detection, enabling better treatment planning and potentially improving survival rates for lung cancer patients.

**PROBLEM STATEMENT**

The early detection of lung cancer remains a challenging problem in the healthcare industry. The primary issue is developing an efficient and reliable prediction model that can analyze clinical data and accurately predict the likelihood of lung cancer. This project seeks to implement machine learning techniques to solve the problem by providing a data-driven approach to lung cancer detection.

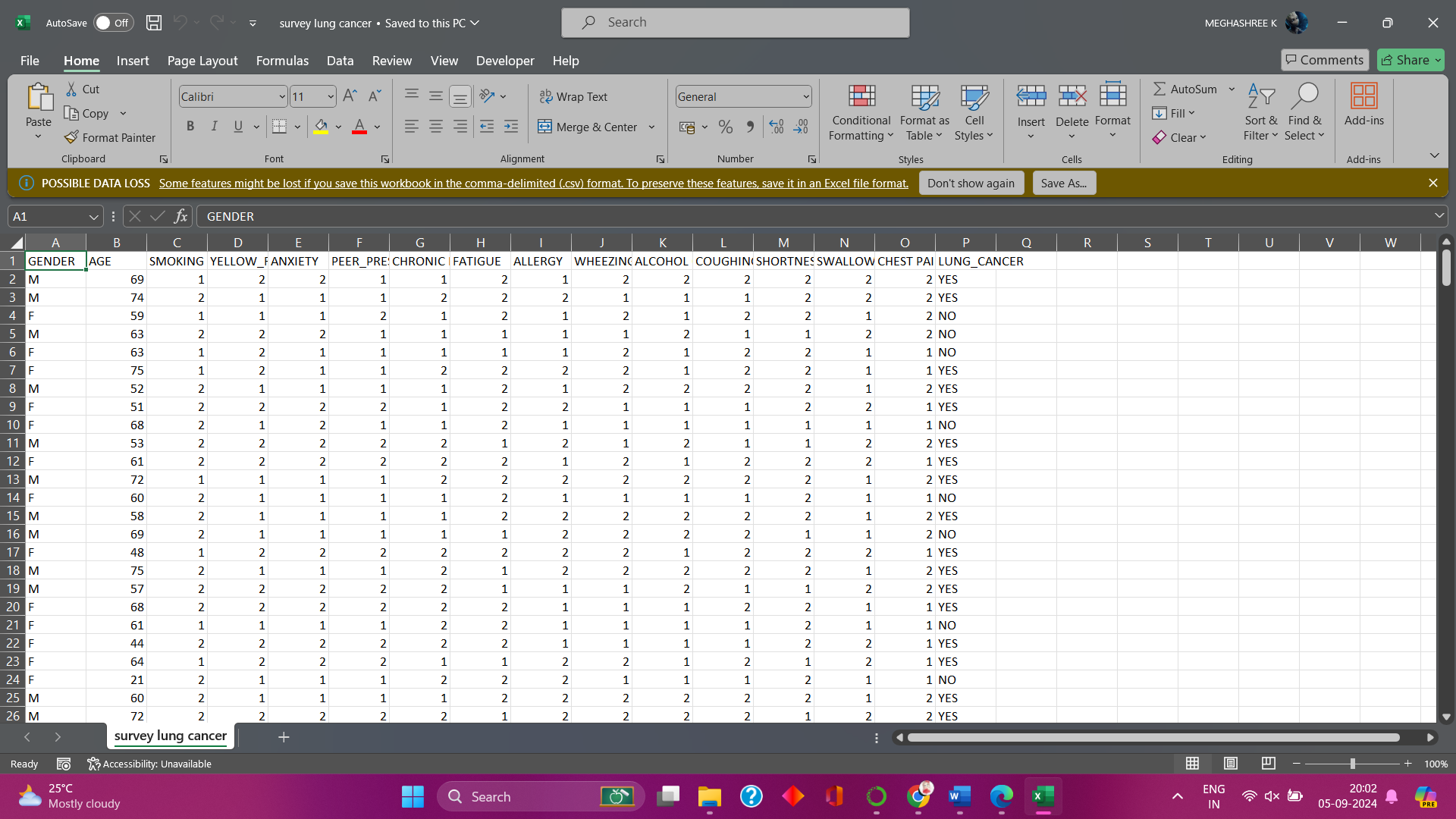
**DATASET**

Here survey lung cancer dataset from Kaggle is used.

The dataset describes factors related to lung cancer, potentially for the purpose of analyzing patterns or risk factors associated with the disease. It includes the following columns:

* **GENDER**: The gender of the individual (e.g., 'M' for male, 'F' for female).
* **AGE**: The age of the individual.
* **SMOKING**: Indicates whether the individual smokes (binary value, 1 for smoker, 2 for non-smoker).
* **YELLOW\_FINGERS**: Represents if the individual has yellow fingers, a potential sign of smoking (coded 1 or 2).
* **ANXIETY**: Indicates the level of anxiety (coded 1 or 2).
* **PEER\_PRESSURE**: Indicates if peer pressure is a factor in the individual's smoking habits (coded 1 or 2).
* **CHRONIC DISEASE**: Indicates if the individual has a chronic disease (coded 1 or 2).
* **FATIGUE**: Measures the individual's level of fatigue (coded 1 or 2).
* **ALLERGY**: Indicates if the individual has allergies (coded 1 or 2).
* **WHEEZING**: Indicates whether the individual experiences wheezing (coded 1 or 2).
* **ALCOHOL CONSUMING**: Indicates if the individual consumes alcohol (coded 1 or 2).
* **COUGHING**: Indicates if the individual experiences coughing (coded 1 or 2).
* **SHORTNESS OF BREATH**: Indicates if the individual has shortness of breath (coded 1 or 2).
* **SWALLOWING DIFFICULTY**: Indicates if the individual has difficulty swallowing (coded 1 or 2).
* **CHEST PAIN**: Indicates if the individual experiences chest pain (coded 1 or 2).
* **LUNG\_CANCER**: The presence or absence of lung cancer ('YES' or 'NO').

This dataset could be used to analyze the correlation between these factors and the likelihood of developing lung cancer.



**FIGURE 1:** Dataset of survey lung cancer

**METHODOLOGY**

**Working Mechanisms:**

1. Loading the Lung Cancer Dataset

The first step is to load the lung cancer dataset. This involves reading the dataset, which contains features related to individual health conditions, lifestyle factors, and lung cancer status (Yes/No). Pandas or another library like NumPy can be used to load the dataset and explore its structure.

1. Data Preprocessing

Preprocessing is essential for preparing the data for model building. It includes:

* **Handling missing values**: Check for missing data and impute or remove it.
* **Encoding categorical variables**: Features like 'GENDER' and 'LUNG\_CANCER' are categorical. They can be encoded using label encoding or one-hot encoding.
* **Feature scaling**: Normalizing or standardizing features like age and other continuous values to bring them to a similar range, making it easier for models to learn.

1. Splitting the Dataset

Once the data is preprocessed, it needs to be split into training and testing sets. The training set is used to build the model, while the testing set helps evaluate its performance.

1. Building the Model

For the lung cancer classification, a machine learning model such as Logistic Regression, Decision Trees, or a Neural Network can be employed. In this example, Logistic Regression is chosen as a simple and interpretable model.

1. Training the Model

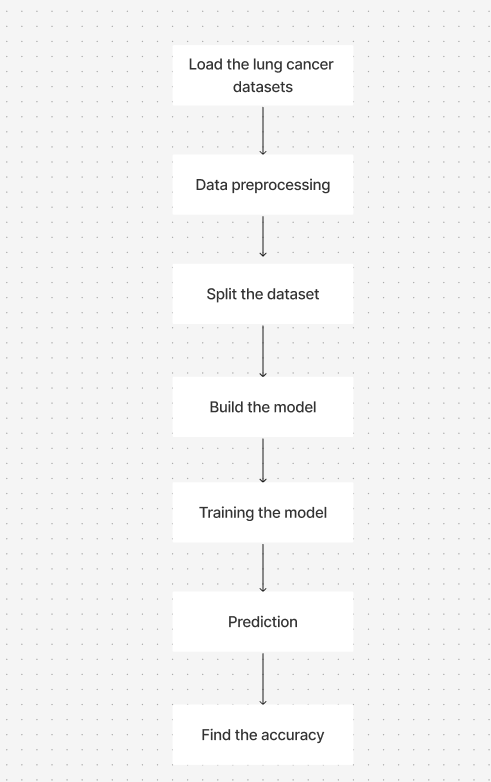
The model is trained on the training data using the fit method, where it learns the relationships between the features and the target variable (lung cancer).

1. Prediction

After training the model, predictions are made on the test set. This helps determine how well the model generalizes to new, unseen data.

1. Find the Accuracy

The model's accuracy is calculated by comparing the predicted values to the actual values from the test set. This is done using accuracy metrics such as accuracy score, precision, recall, or the confusion matrix.



**FIGURE 2:** Represents the working mechanisms

**REQUIRED ALGORITHMS**

* **Libraries Used**:
  + Pandas for data manipulation
  + NumPy for numerical operations
  + Matplotlib and Seaborn for data visualization
  + Scikit-learn for model implementation and evaluation
* **Algorithms**:

**Logistic Regression** is a simple yet powerful supervised machine learning algorithm used for binary classification tasks. Instead of predicting continuous values like linear regression, it predicts the probability that a given input belongs to one of two classes (e.g., "yes" or "no," "0" or "1"). It applies the **logistic function** (also called the sigmoid function) to map predicted values to probabilities between 0 and 1.

The model outputs probabilities, and a threshold (usually 0.5) is applied to decide the final class. Logistic regression is widely used in medical fields, such as disease prediction, due to its simplicity and interpretability.

**Linear Regression** is a supervised machine learning algorithm used for predicting continuous values. It models the relationship between an independent variable (or multiple variables) and a dependent variable by fitting a straight line to the data.Linear regression is used in scenarios like predicting house prices, sales forecasting, and more, where the goal is to understand the linear relationship between variables.

**Evaluation Metrics**:

1. **Accuracy Score**:

Measures the proportion of correctly predicted instances out of the total. It's mainly used for classification tasks.

1. **Confusion Matrix**:

A table used to evaluate the performance of a classification model by showing true positives (TP), true negatives (TN), false positives (FP), and false negatives (FN). It helps assess how well the model is performing across different classes.

1. **Mean Squared Error (MSE)**:

A common evaluation metric for regression models that measures the average squared difference between actual and predicted values. Lower MSE indicates a better model.

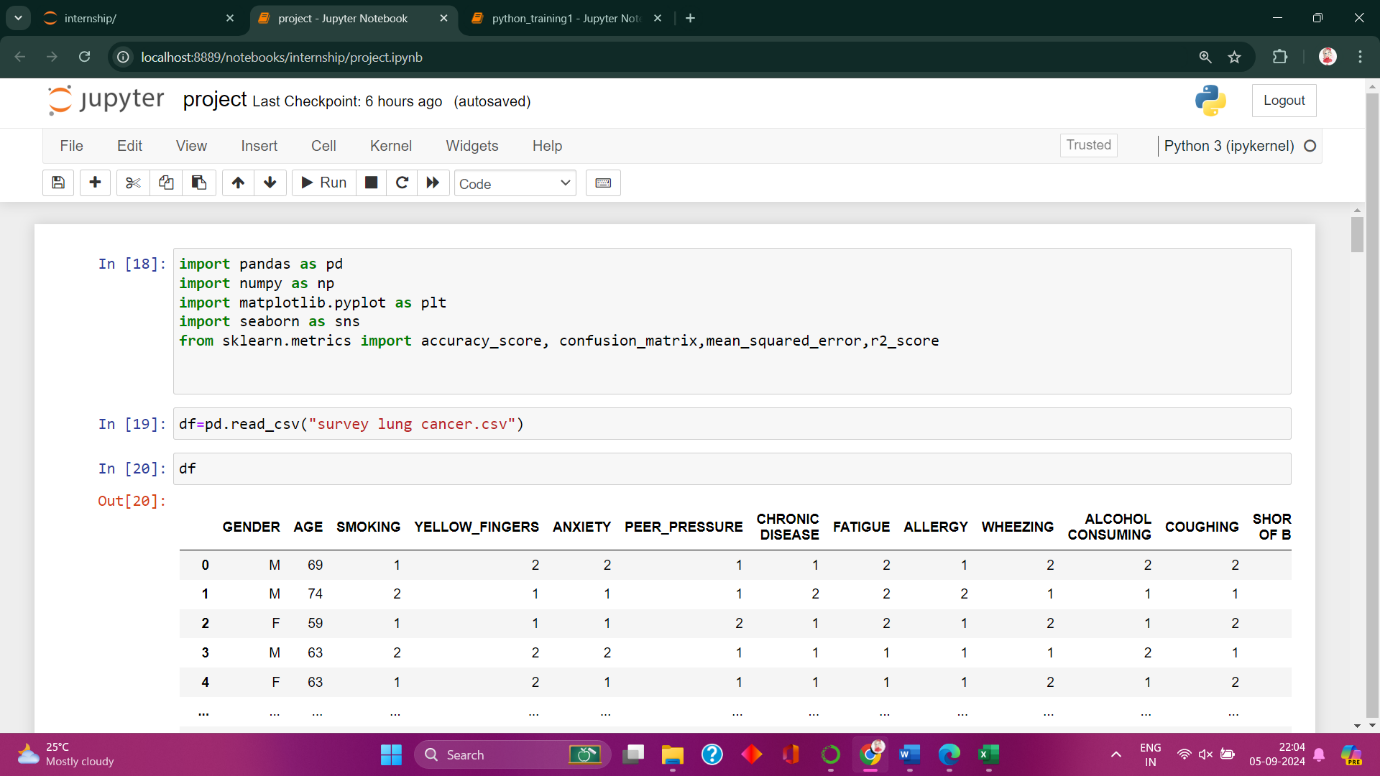
1. **R² Score:**

Indicates how well the independent variables explain the variance in the dependent variable in regression models. R² values range from 0 to 1, with higher values indicating a better fit.

**IMPLEMENTATION**

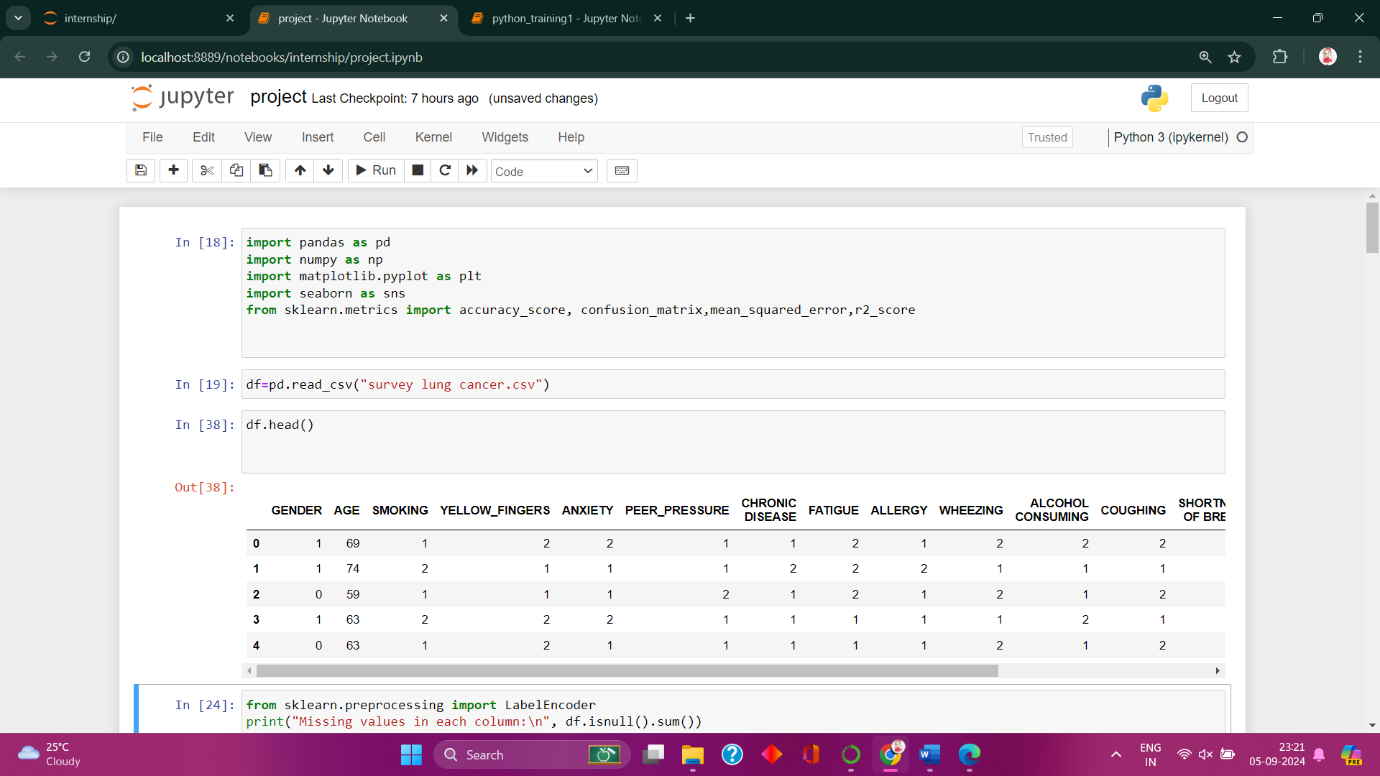
Implementation is a word used to describe interaction of elements in programming language. In computer programming, a programming language implementation is a system of executing computer programs.

**IMPORT LIBRARIES**



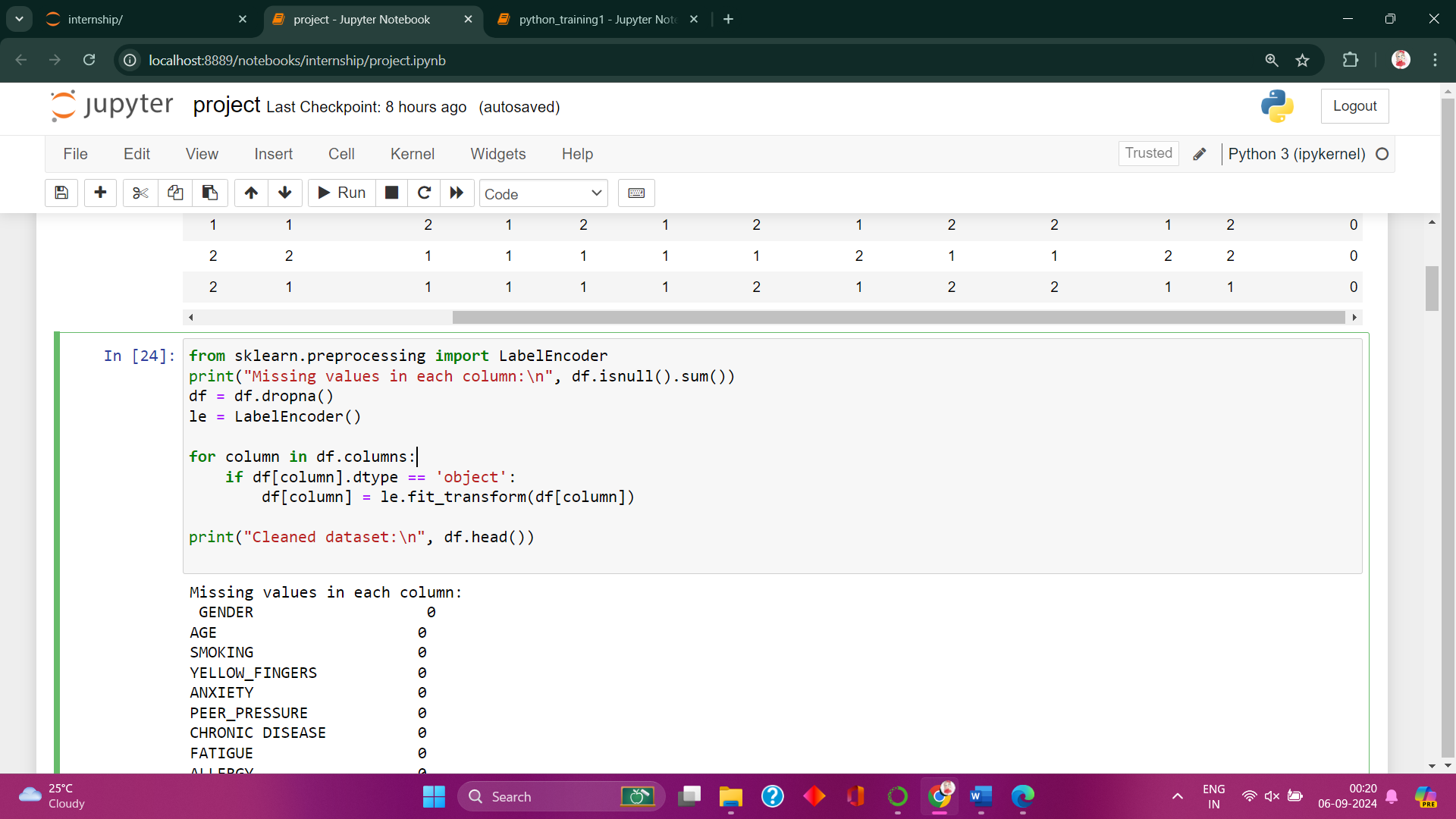
To work with data and visualize results in Python, import the following libraries and metrics: **Numpy** for array operations (short for "Numeric Python"), **Pandas** for data manipulation (from "panel data"), and **Matplotlib** for plotting. Metrics include **Accuracy Score** (the ratio of correctly predicted to total instances), **Confusion Matrix** (a table comparing predicted and actual values), **Mean Squared Error (MSE)** (the average squared difference between predicted and actual values), and **R-squared (R²)** (the proportion of variance in the dependent variable predictable from independent variables).

**UPLOAD THE DATASET**

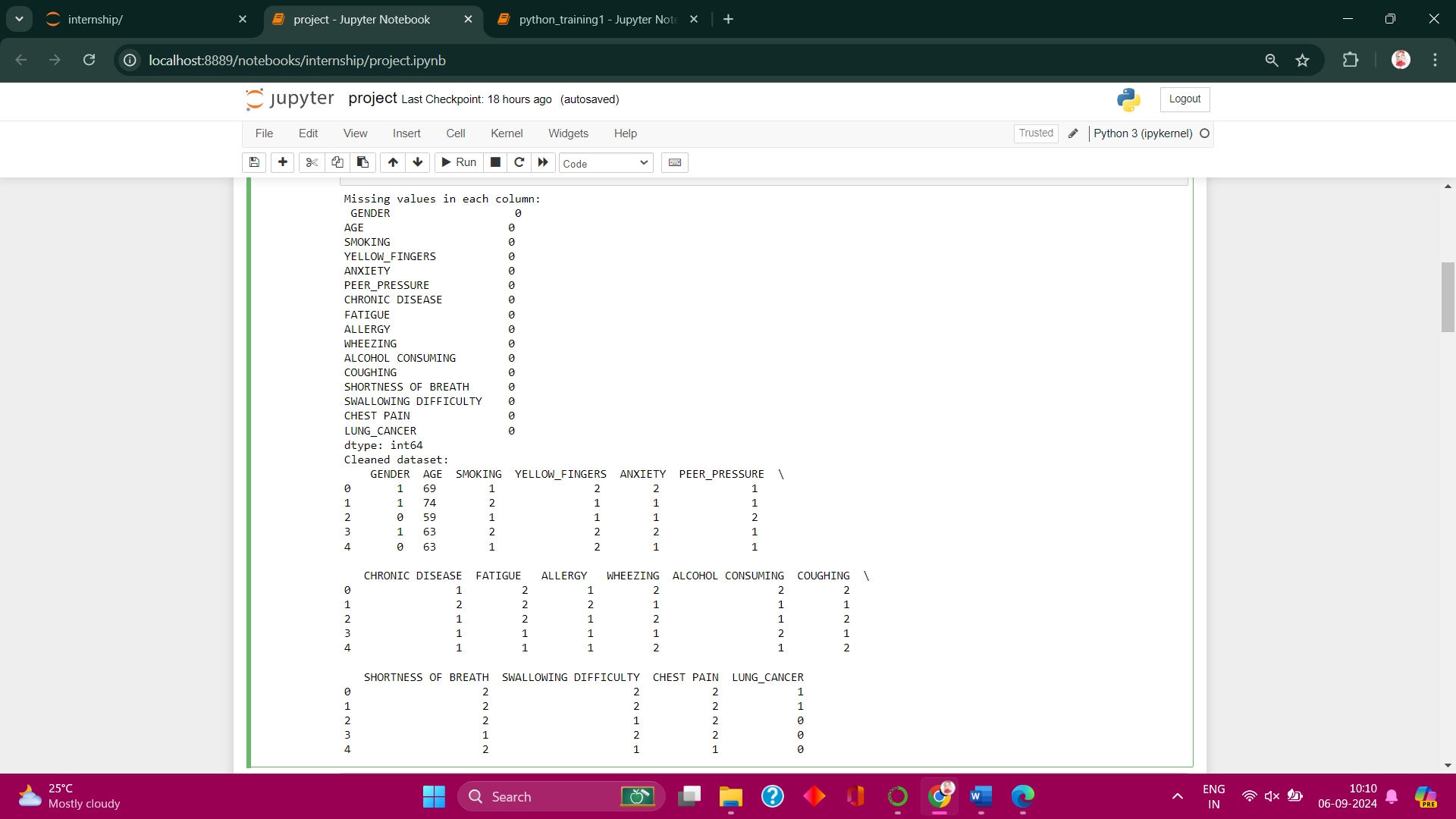
****

Upload the survey lung cancer dataset, from Kaggle.

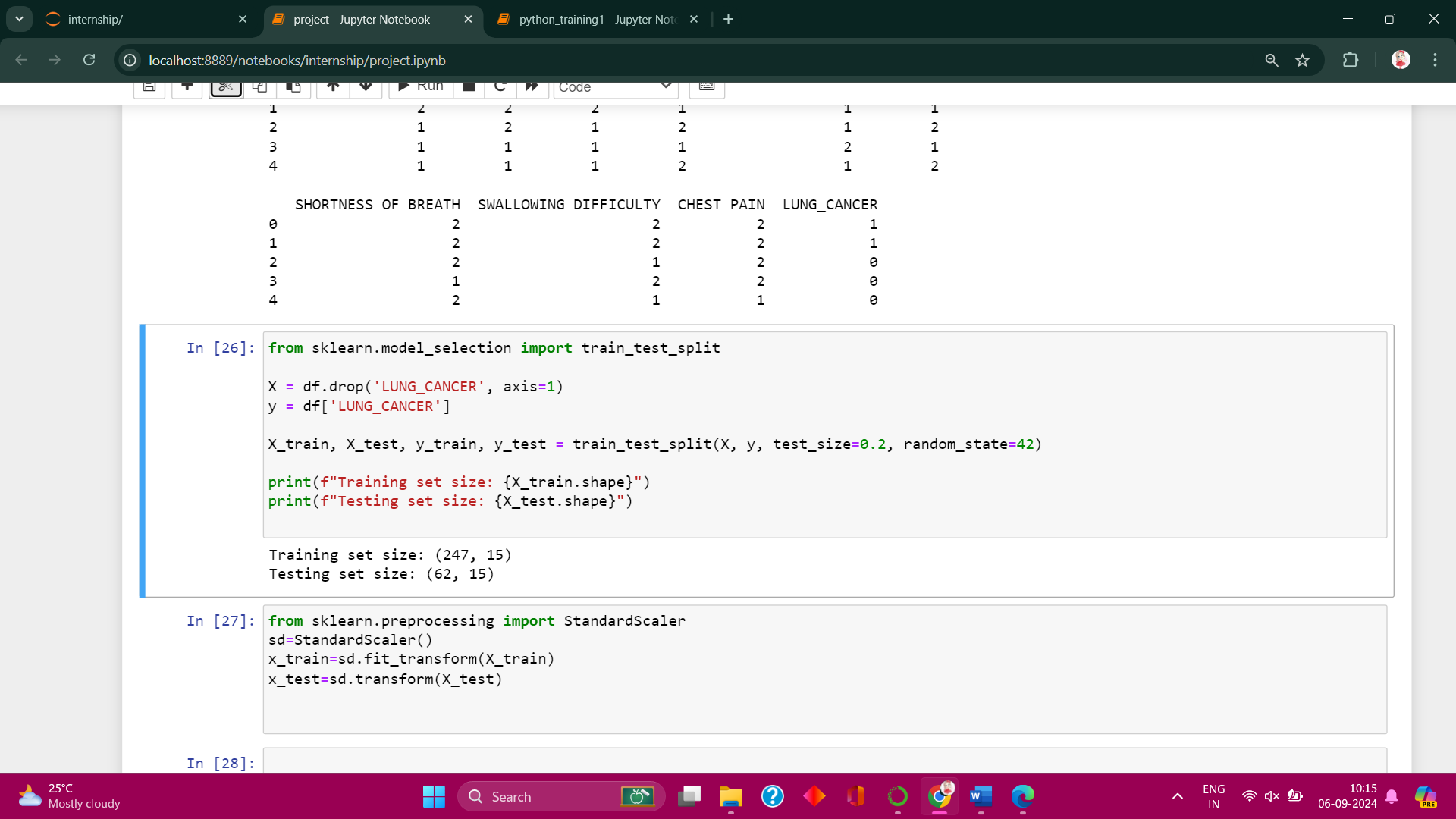
**DATA PREPROCESSING**

****

The code removes rows with missing values, which is part of data cleaning and function **df.isnull().sum()** checks for missing values in each column of the DataFrame.The line **df = df.dropna()** removes any rows from the DataFrame that contain missing values. If the column contains categorical data, it applies **LabelEncoder** to transform these categorical values into numerical labels. Data preprocessing and cleaning steps ensure that your dataset is in a suitable format for further analysis or machine learning tasks.



**SPLITTING THE DATA**

****

Import the **train\_test\_split** function from the **sklearn.model\_selection** module, part of the scikit-learn library. This function is used to split datasets into training and testing sets.

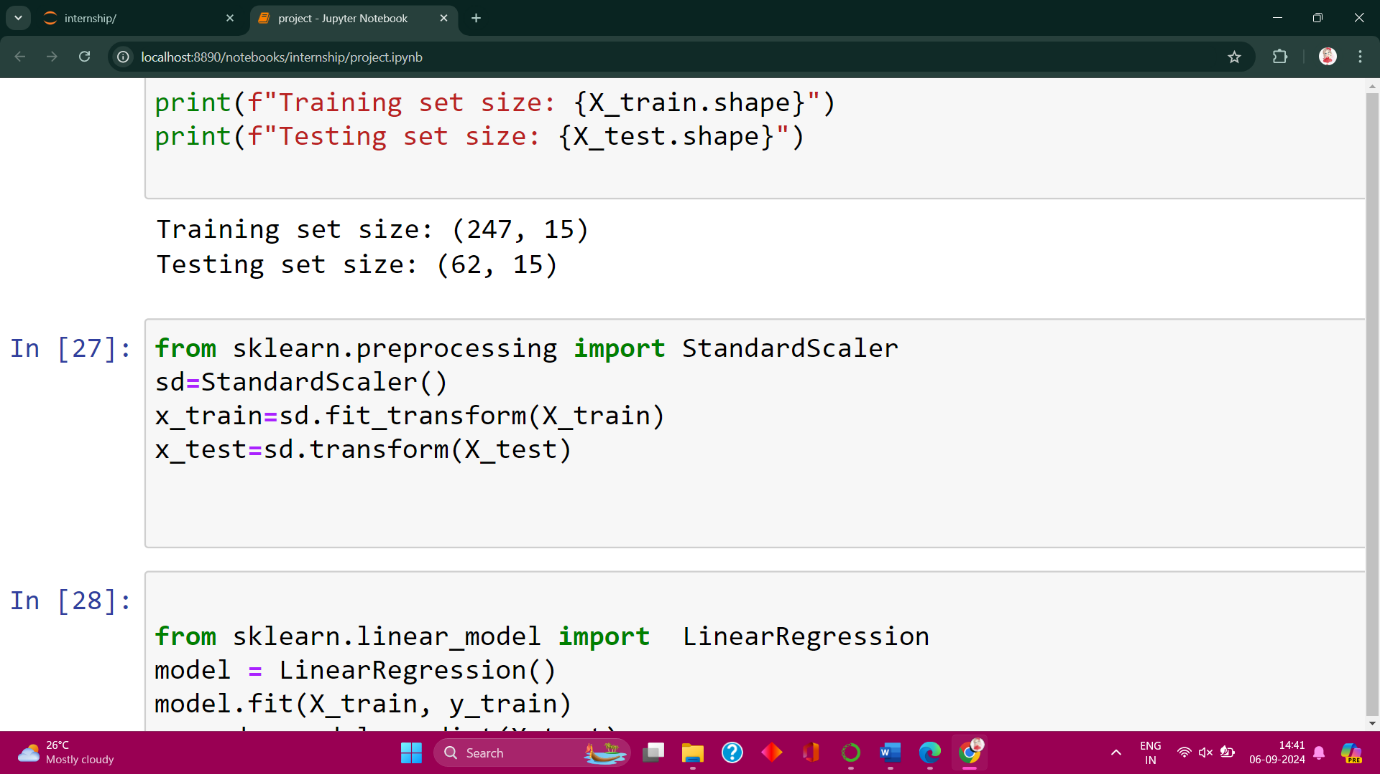
**X:** Contains all the features that are independent variables except the target column **LUNG\_CANCER**, **y:** Contains only the target that are dependent variable, which is **LUNG\_CANCER.**

**X\_train** and **y\_train**: 80% of the data (default, test\_size=0.2) is used for training.

**X\_test** and **y\_test**: 20% of the data is reserved for testing the model’s performance.

**random\_state**=42 ensures that the split is reproducible. **shape** is that number of rows and columns are printed. Splitting the data ensures that the model is trained on one part of the data, **X\_train** and **y\_train**.The model is evaluated on unseen data,**X\_test** and **y\_test**, which helps check how well the model generalizes to new, unseen examples.

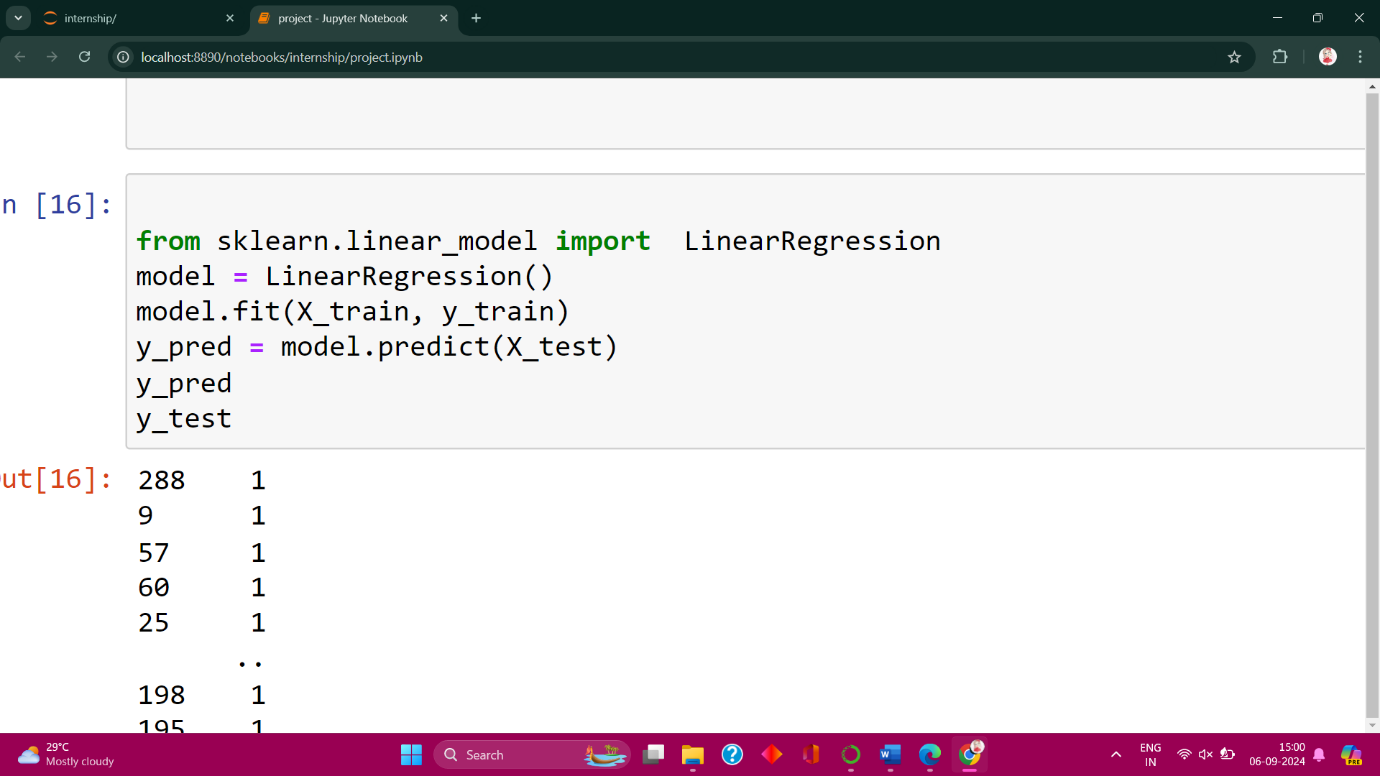
**FEARURE SCALING**

****

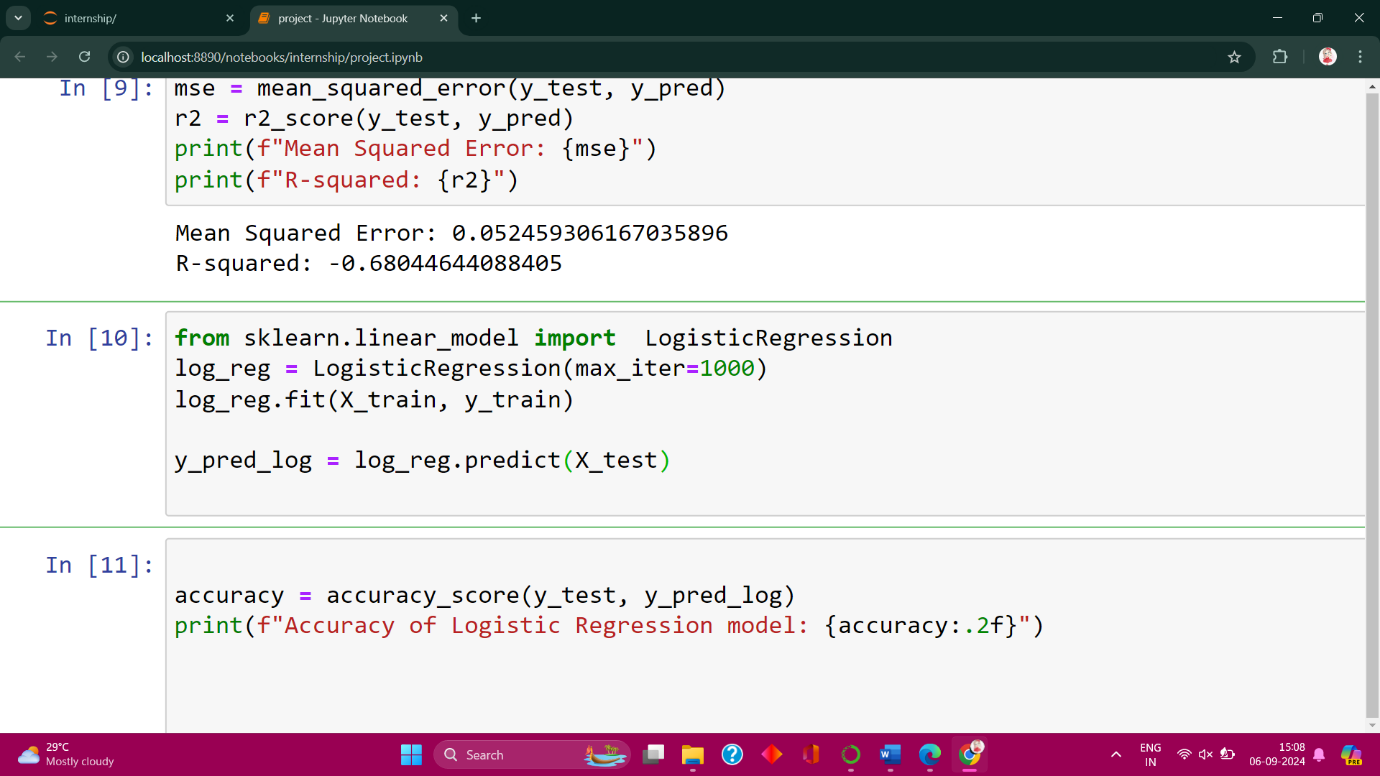
This code applies standardization to the training and testing datasets using the **StandardScaler** from **sklearn.preprocessing**.This imports the **StandardScaler** class, which is used to scale features so they have a mean of 0 and a standard deviation of 1.

Initialize the Scaler,then **fit\_transform** calculates the mean and standard deviation of the training data and then transforms the data by scaling it, **transform** uses the mean and standard deviation calculated from the training data and applies the same transformation to the testing data.

**TRAINING AND PREDICTION**

****

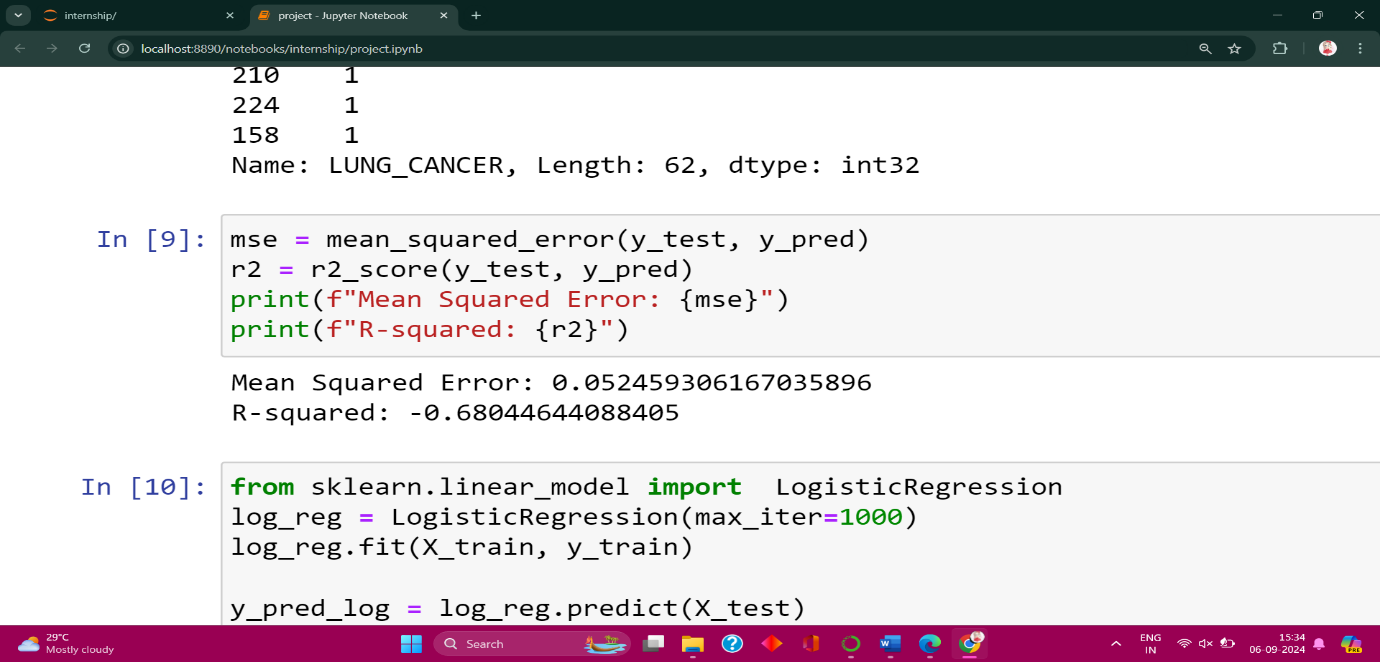
In the code, **Linear Regression** is used to model the relationship between features (X\_train) and the target (y\_train). The model is trained using the fit() function, learning the coefficients for the best-fit line. It then predicts the target values (y\_pred) for the test data (X\_test), allowing comparison with actual values (y\_test).



In the code, **Logistic Regression** is used to model the probability of a binary outcome (like classifying lung cancer). The model is trained using fit() on the training data (X\_train, y\_train), where it learns the relationship between features and the target. It then predicts the class labels (y\_pred\_log) for the test data (X\_test), useful for classification tasks.

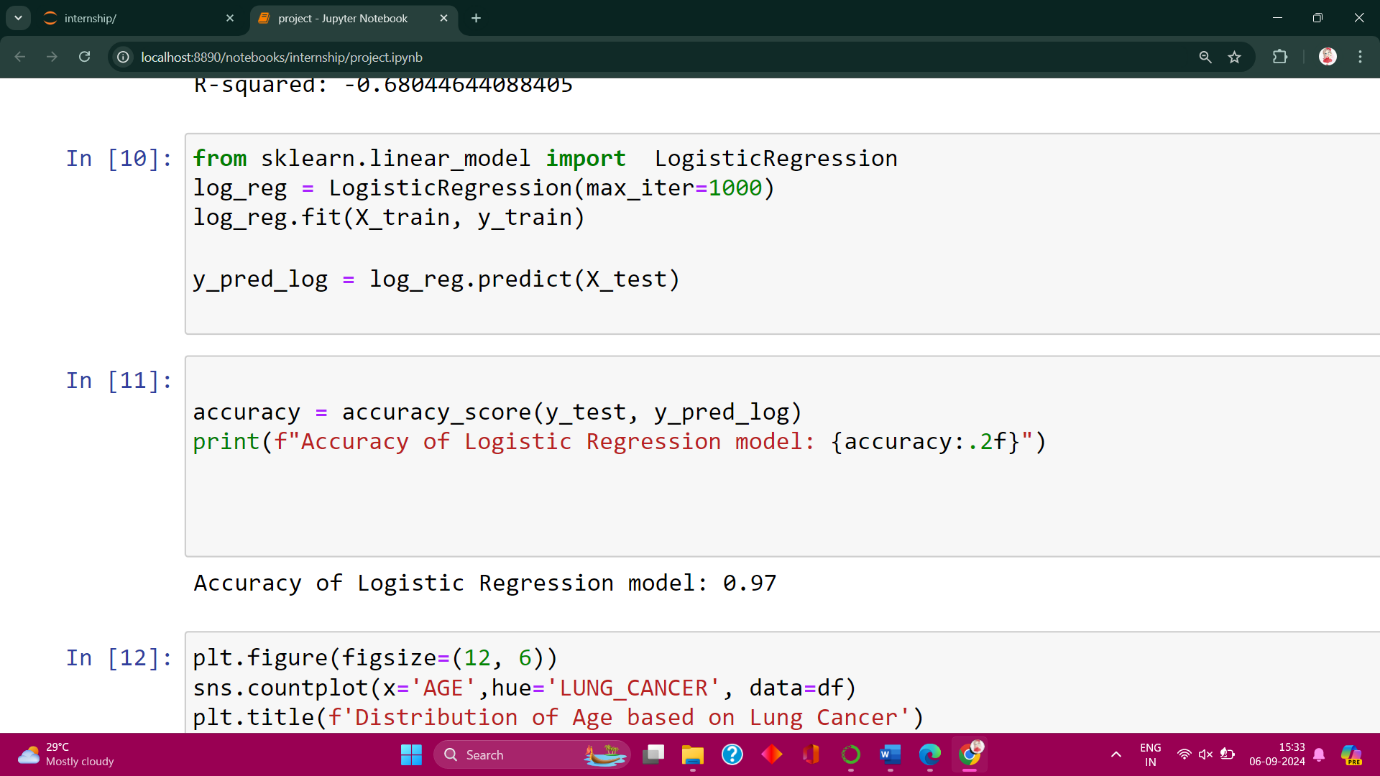
**RESULTS AND DISCCUSSION**

**RESULT OF LINEAR REGRESSION**



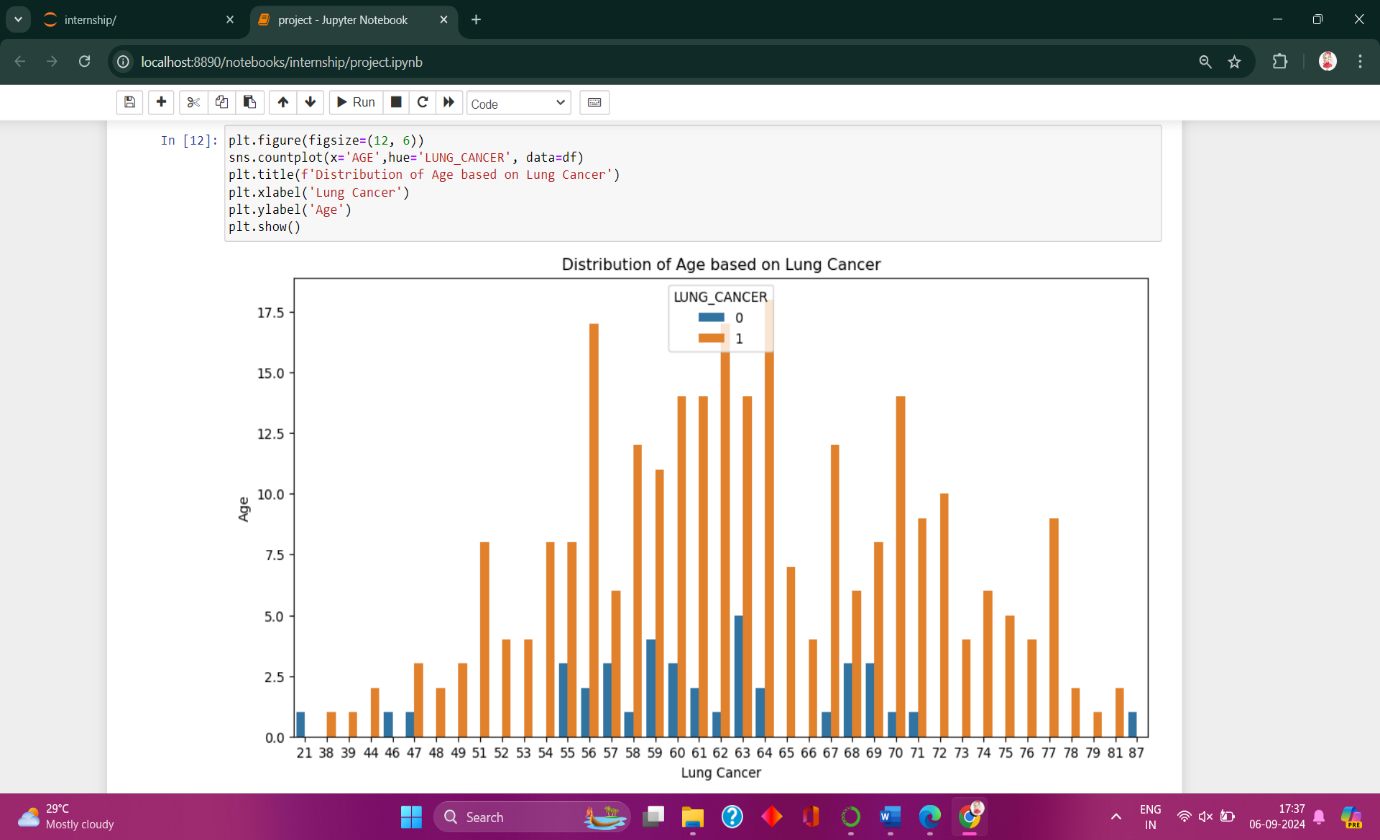
The Mean Squared Error (MSE) is 0.052, indicating relatively small differences between actual and predicted values. However, the R-squared (R²) is negative (-0.68), which shows the model is performing poorly and not fitting the data well. A negative R² suggests that the model is worse than a baseline prediction using the mean. Overall, the model's performance is unsatisfactory despite the low MSE.

**ACCURACY OF LOGISTIC REGRESSION**



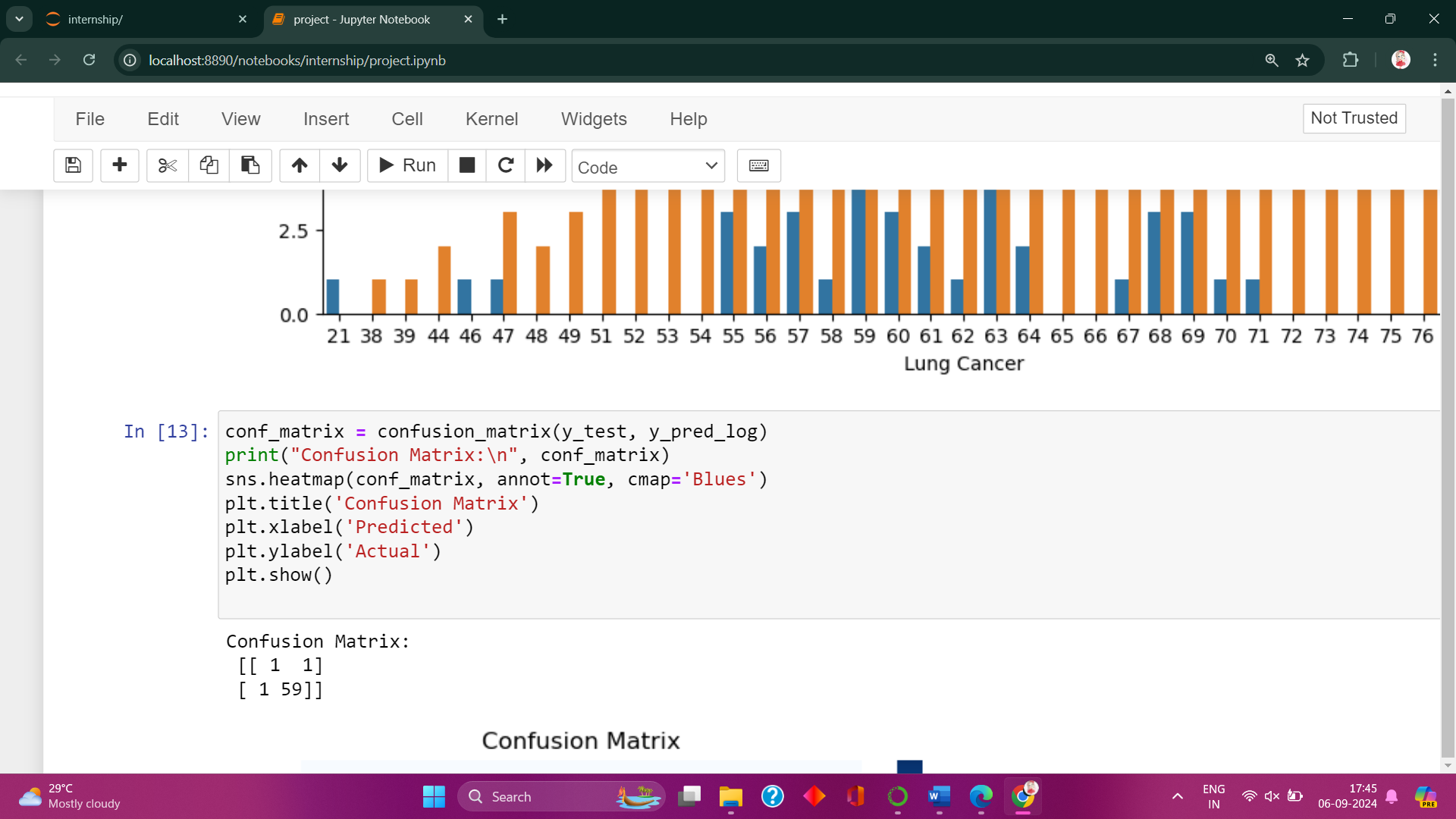
The code calculates the accuracy of the Logistic Regression model, which is **0.97** (or 97%). This means the model correctly classified 97% of the test data, indicating a high level of performance for this classification task.

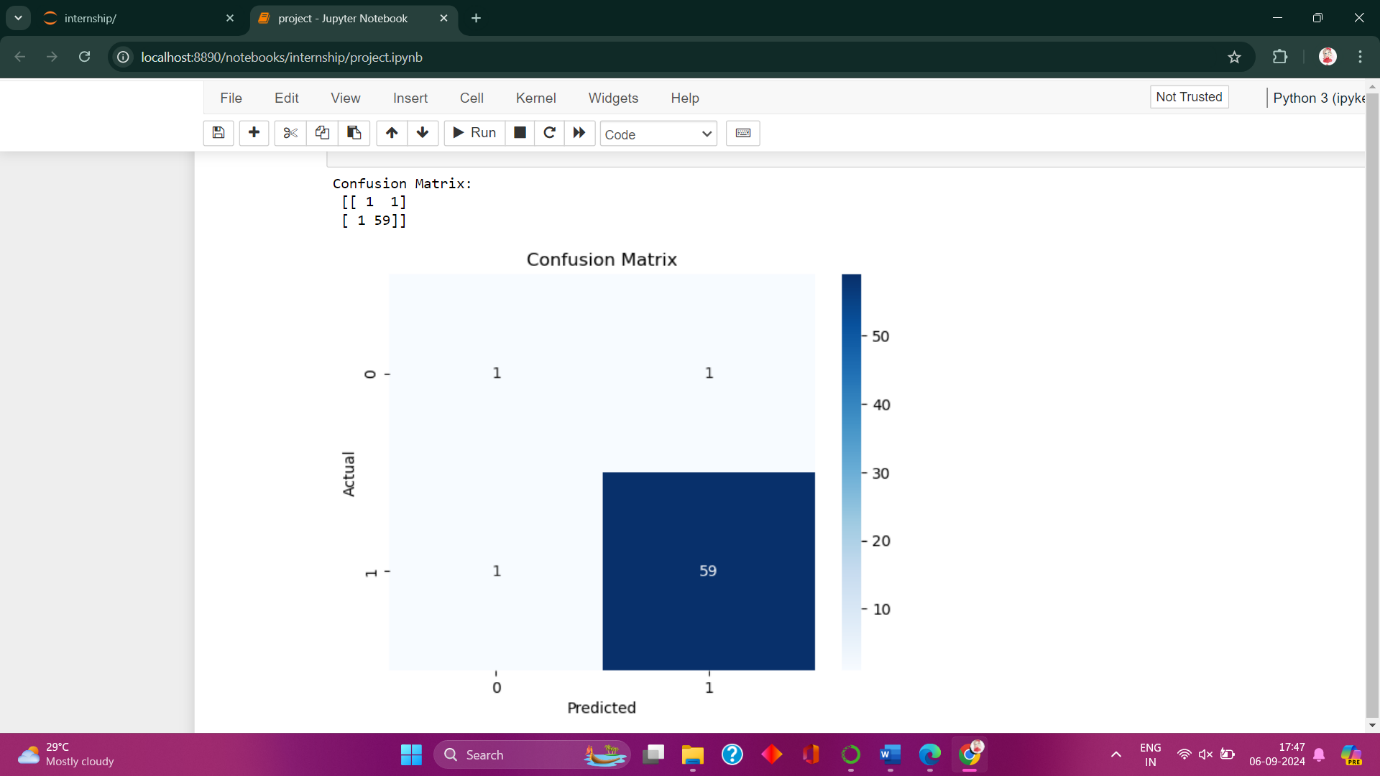
**GRAPHICAL REPRESENTATION**



The code generates a count plot that visualizes the distribution of ages among individuals based on their lung cancer status. The x-axis represents different age groups, while the y-axis shows the count of individuals. The plot uses two colors: **orange** (representing individuals with lung cancer, labeled as "1") and **blue** (representing individuals without lung cancer, labeled as "0"). This helps in identifying which age groups have more cases of lung cancer, with a noticeable concentration of lung cancer cases in the age range of 55 to 65. The plot provides insights into the correlation between age and lung cancer prevalence.

**CONFUSION MATRIX**

****

****

The confusion matrix shows the performance of the Logistic Regression model in predicting lung cancer. It correctly predicted 59 cases of lung cancer (True Positives) and 1 case where the person does not have lung cancer (True Negative). However, it misclassified 1 case of lung cancer as non-cancer (False Negative) and 1 case of non-cancer as lung cancer (False Positive). Overall, the model performs well with only two misclassifications, mainly in identifying non-cancer cases.

**CONCLUSION**

In this project, the analysis demonstrates that Logistic Regression is an effective model for predicting lung cancer, achieving a high accuracy of 97%. By contrast, the Linear Regression model performs poorly, as indicated by its low R-squared value and high mean squared error, making it unsuitable for this classification task. The confusion matrix further confirms that the Logistic Regression model has strong predictive power, with minimal misclassification of lung cancer cases. The data visualization also highlights the relationship between age and lung cancer, offering valuable insights into how the disease distribution varies with age. Overall, the preprocessing steps and model evaluation show that Logistic Regression is a suitable choice for this type of binary classification problem.

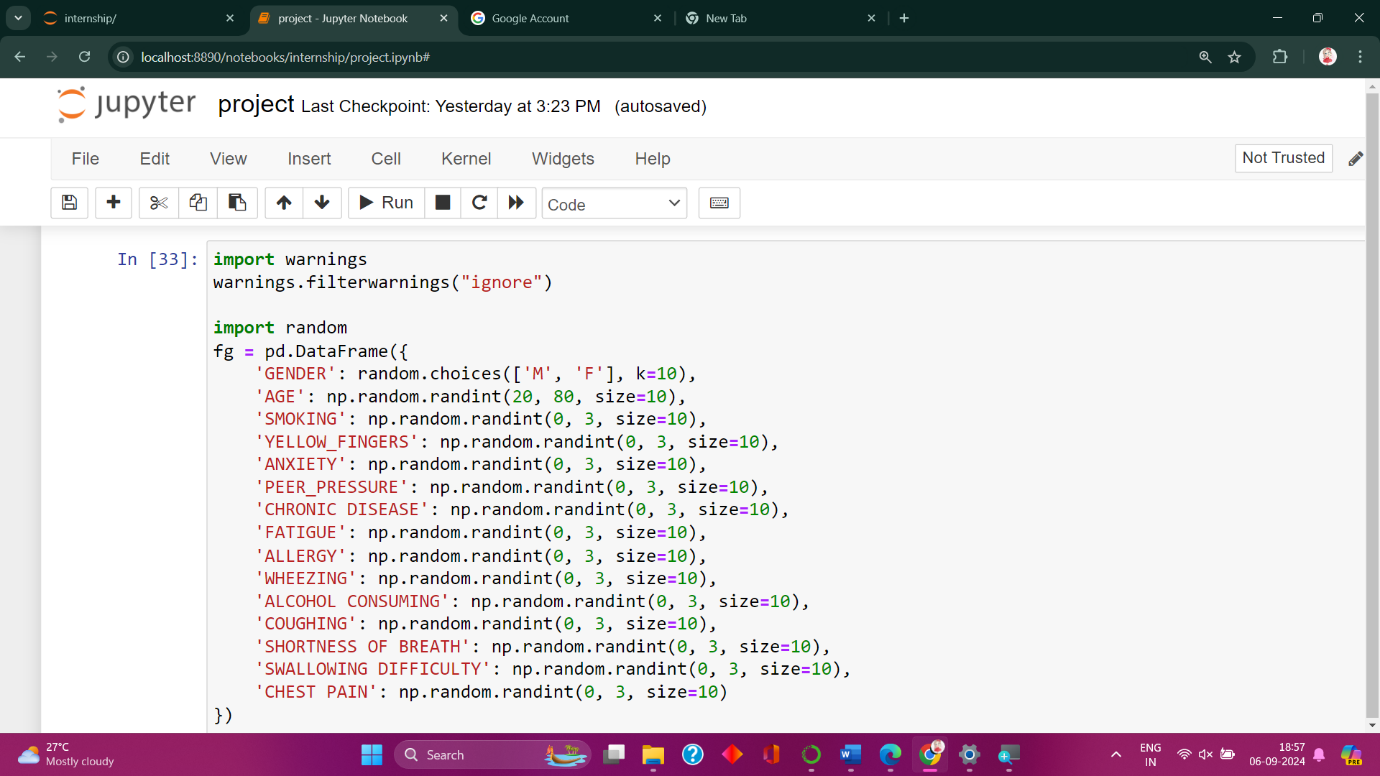
**SCOPE FOR FUTURE ENHANCEMENT**

This can be incredibly helpful in predicting outcomes for new data. By leveraging the insights gained from training on existing data, you can use the model to predict the likelihood of lung cancer for new patients based on their characteristics, such as age, symptoms, and medical history. This enables healthcare professionals to make informed decisions about diagnosis, treatment, and prevention.

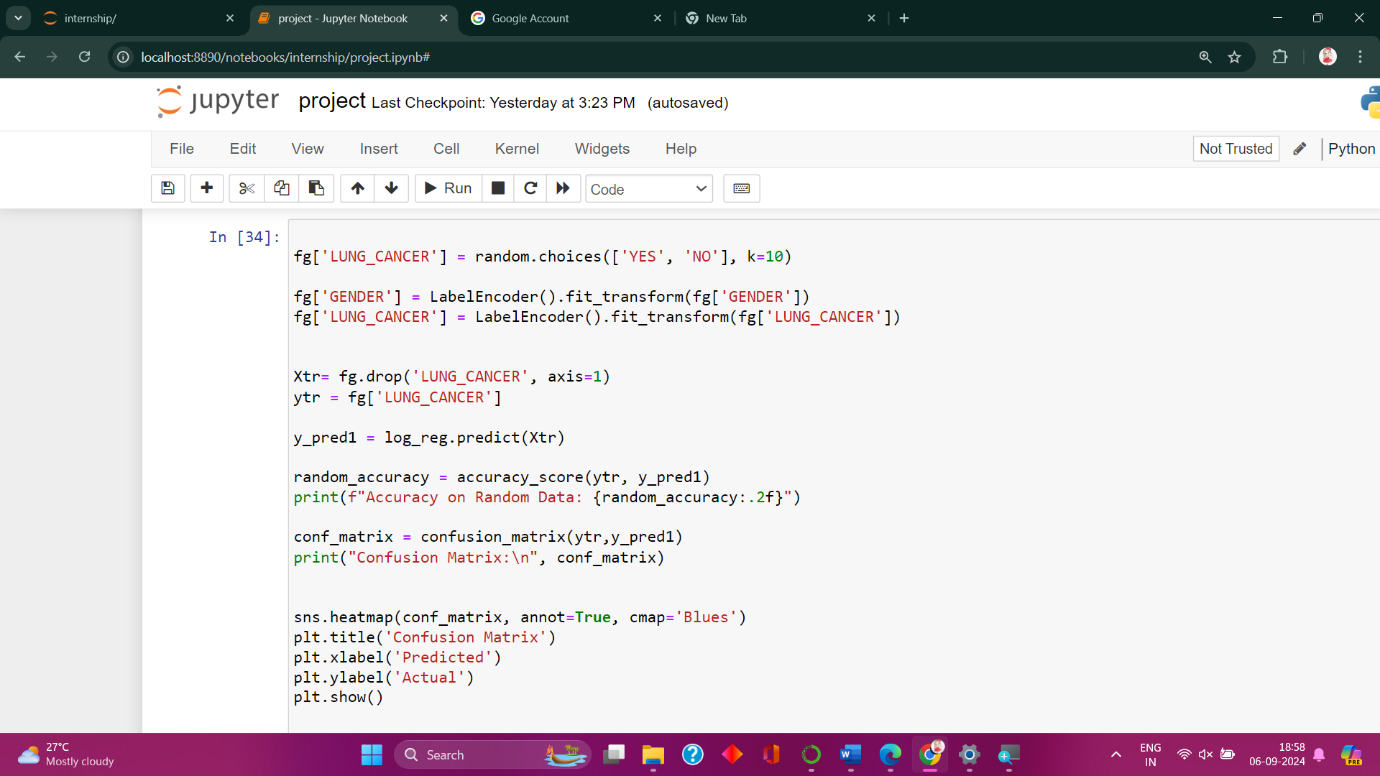
Additionally, the model can predict the risk of developing lung cancer for individuals with certain risk factors, allowing for early intervention and targeted interventions. The model can also predict treatment outcomes, helping doctors choose the most effective course of action for new patients.

Furthermore, it can predict survival rates, providing valuable information for patients and healthcare providers. By identifying high-risk groups or populations, the model can also inform public health initiatives and resource allocation. Overall, your model has the potential to make a significant impact on lung cancer diagnosis, treatment, and prevention.

**PREDICTING FOR NEW DATA**

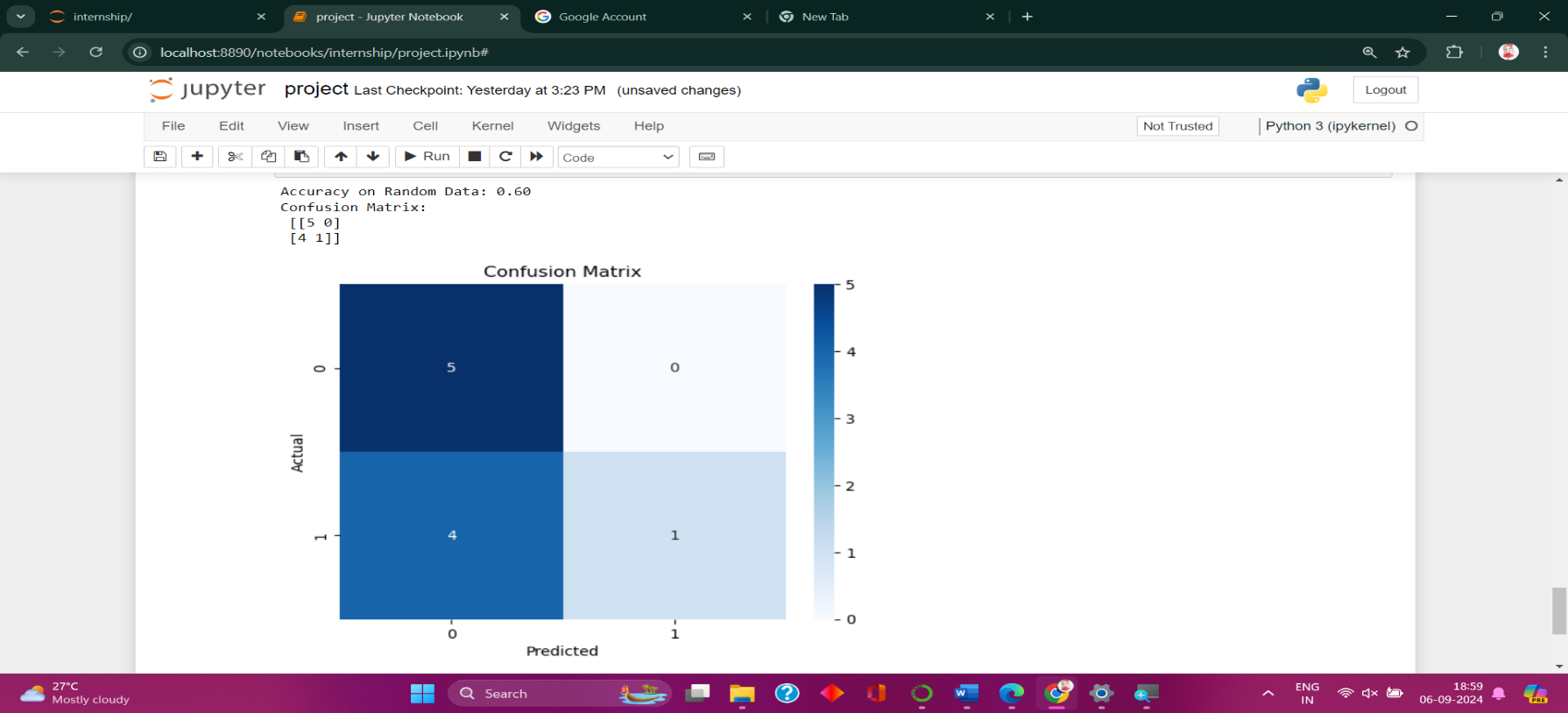
****

Here **DataFrame** is created and named as **fg** using the **pandas** library, with 10 rows and several columns representing various health and lifestyle attributes. It imports the **warnings** and **random** modules, with warnings suppressed for cleaner output. The **DataFrame** includes columns such as 'GENDER', 'AGE', 'SMOKING', 'YELLOW\_FINGERS', and others. The **'GENDER'** column contains random selections of **'M'** (Male) or **'F'** (Female). The **'AGE'** column has random integers between 20 and 80. The other columns, which represent different health conditions and lifestyle factors, are filled with random integers between 0 and 2, indicating various levels or categories for each attribute.



The code snippet updates the **fg DataFrame** by adding a new column, **'LUNG\_CANCER'**, with random values of **'YES'** or **'NO**', and then encodes the 'GENDER' and 'LUNG\_CANCER' columns into numerical values using LabelEncoder. It separates the features **Xtr** from the target variable **ytr** and uses a logistic regression model **log\_reg** to make predictions on the feature set. The accuracy of these predictions is calculated and printed.

Additionally, a confusion matrix is generated to evaluate the performance, which is then visualized using a heatmap with **seaborn**. This provides a graphical representation of the prediction results against the actual values.



The accuracy score of 0.60 means the model correctly predicted the class 60% of the time. The confusion matrix reveals that the model correctly identified 5 cases as 'NO' (true negatives) and 1 case as 'YES' (true positive). It incorrectly classified 4 cases as 'NO' when they were actually 'YES' (false negatives), and made no incorrect 'YES' predictions for 'NO' cases (false positives).

**APPENDIX**

**import pandas as pd**

**import numpy as np**

**import matplotlib.pyplot as plt**

**import seaborn as sns**

**from sklearn.metrics import accuracy\_score, confusion\_matrix,mean\_squared\_error,r2\_score**

**df=pd.read\_csv("survey lung cancer.csv")**

**df.head()**

**from sklearn.preprocessing import LabelEncoder**

**print("Missing values in each column:\n", df.isnull().sum())**

**df = df.dropna()**

**le = LabelEncoder()**

**for column in df.columns:**

**if df[column].dtype == 'object':**

**df[column] = le.fit\_transform(df[column])**

**print("Cleaned dataset:\n", df.head())**

**from sklearn.model\_selection import train\_test\_split**

**X = df.drop('LUNG\_CANCER', axis=1)**

**y = df['LUNG\_CANCER']**

**X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2, random\_state=42)**

**print(f"Training set size: {X\_train.shape}")**

**print(f"Testing set size: {X\_test.shape}")**

**from sklearn.preprocessing import StandardScaler**

**sd=StandardScaler()**

**x\_train=sd.fit\_transform(X\_train)**

**x\_test=sd.transform(X\_test)**

**from sklearn.linear\_model import LinearRegression**

**model = LinearRegression()**

**model.fit(X\_train, y\_train)**

**y\_pred = model.predict(X\_test)**

**y\_pred**

**y\_test**

**mse = mean\_squared\_error(y\_test, y\_pred)**

**r2 = r2\_score(y\_test, y\_pred)**

**print(f"Mean Squared Error: {mse}")**

**print(f"R-squared: {r2}")**

**from sklearn.linear\_model import LogisticRegression**

**log\_reg = LogisticRegression(max\_iter=1000)**

**log\_reg.fit(X\_train, y\_train)**

**y\_pred\_log = log\_reg.predict(X\_test)**

**accuracy = accuracy\_score(y\_test, y\_pred\_log)**

**print(f"Accuracy of Logistic Regression model: {accuracy:.2f}")**

**plt.figure(figsize=(12, 6))**

**sns.countplot(x='AGE',hue='LUNG\_CANCER', data=df)**

**plt.title(f'Distribution of Age based on Lung Cancer')**

**plt.xlabel('Lung Cancer')**

**plt.ylabel('Age')**

**plt.show()**

**conf\_matrix = confusion\_matrix(y\_test, y\_pred\_log)**

**print("Confusion Matrix:\n", conf\_matrix)**

**sns.heatmap(conf\_matrix, annot=True, cmap='Blues')**

**plt.title('Confusion Matrix')**

**plt.xlabel('Predicted')**

**plt.ylabel('Actual')**

**plt.show()**

**BIBILOGRAPHY**

<https://intellipaat.com/blog/what-is-logistic-regression/>

<https://scikit-learn.org/stable/index.html>

<https://www.kaggle.com/code/sandragracenelson/lung-cancer-prediction>

<https://www.kaggle.com/datasets/jillanisofttech/lung-cancer-detection>