

# LUNG CANCER DETECTION

Gnana Deepak Madduri, Ifrat Zaman, Shubham Pawar

*College of Engineering and Applied Sciences, Western Michigan University*

gnanadeepak.madduri@wmich.edu

ifrat.zaman@wmich.edu

shubhamanil.pawar@wmich.edu

**Abstract** - Lung cancer is the leading cause of death among all cancer patients. Like all other cancers, it is difficult, and expensive, to detect and diagnose lung cancer. However, one of the factors that increases the chance of a full recovery from lung cancer is the detection of the disease in its preliminary stages. With the advancements in science and technology, there are Machine Learning models that can now detect cancer. Three such ML models are trained and tested in this study to evaluate which model yields the best results for lung cancer prediction. The Support Vector Model returned the best output with a precision of 1.00, sensitivity of 0.90, and F1-score of 95%. This model was then used to create a ML application that could predict the risk of lung cancer based on just 15 questions. Upon completion of developing the application on local machines, the application was then implemented and run on Microsoft Azure Automation cloud service.

## I. INTRODUCTION

Lung cancer is one of the most fatal forms of cancer worldwide. It is estimated that 1 in every 16 people will suffer from lung cancer in their lifetime. According to a research conducted by the Lung Cancer Research Foundation, it is estimated that 236,740 people will be diagnosed with lung cancer in 2022 in the U.S. [5]. With so many people's lives at stake, it is imperative to be able to detect lung cancer in its early stages in order to diagnose, treat, and cure the disease. This project does exactly that by creating an application that can predict the risk of lung cancer based on just 15 questions.

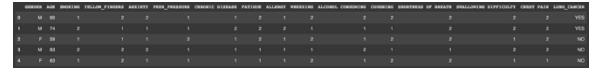
Various studies have shown that smoking is the leading cause of lung cancer and is responsible for 80% of such cases. However, there are several other risk factors that contribute to this deadly lung disease - secondhand smoke, asbestos, air pollution, and automobile exhaust to name a few [5]. In this project, a dataset is obtained from Kaggle containing multiple such risk factors that can be used to predict the risk of lung cancer [1]. Several Machine Learning (ML) models will be implemented and evaluated to identify the best model that predicts lung cancer based on the attributes from the dataset.

### A. Project Objectives

The objective of this project is to create a ML application that can predict the risk of lung cancer. The first half of the project is dedicated to identifying and choosing the best ML model with the best precision and sensitivity to implement into the application. The application will consist of 15 questions that a user will answer in order to find out the risk of having lung cancer.

### B. Dataset Description

The dataset for this project is obtained from Kaggle [1]. There are a total of 309 observations in this structured dataset. Each observation is self-reported data indicating various personal attributes like smoking, anxiety, fatigue, allergy, etc. These attributes are collected in 0s and 1s, where 0s indicate "NO" and 1s indicate "YES" to the respective attributes. The last column of the dataset, LUNG\_CANCER, is the column that contains the information of whether or not the subject of that observation had reported to have lung cancer. All the data is stored in a comma-separated values (.csv) file. The best feature of this dataset is that it contains no missing values. Therefore, the veracity of data is high and analysis outputs will be accurate and useful.



	gender	age	smoking_status	tobacco_consumption	anxiety	stress	fatigue	allergy	asthma	chronic_cough	wheezing	chest_pain	shortness_of_breath	weight_loss	lung_cancer
1	M	75	1	1	1	1	1	1	1	1	1	1	1	1	YES
2	F	65	1	1	1	1	1	1	1	1	1	1	1	1	NO
3	M	45	1	1	1	1	1	1	1	1	1	1	1	1	NO
4	F	55	1	1	1	1	1	1	1	1	1	1	1	1	NO
5	M	70	1	1	1	1	1	1	1	1	1	1	1	1	NO

Figure 1: First 5 observations in the dataset

## II. METHODOLOGY

This section will explain the data cleaning, preprocessing, exploration, implementation and evaluation of three ML models that predict lung cancer using the dataset explained above.

### A. Architecture & Diagram

In order to choose the best ML model for the Lung Cancer Detection application, it is necessary to follow the traditional process of data cleaning, exploration, implementing various ML models, and ML models evaluations. The following diagram is a summary of the methodology used to choose the best ML model for our application.

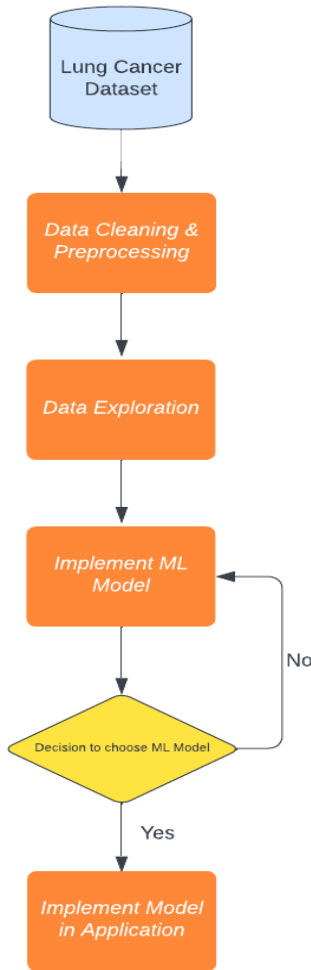


Figure 2: Summary of methodology to choose ML model

### B. Data Cleaning & Preprocessing

The dataset returned no missing value upon calling the `df.isnull().sum()` function. However, upon checking for duplicate rows, the function `df.duplicated().sum()` returned 33, which means there exists 33 duplicate rows. These rows were removed from the dataframe using `df.drop_duplicates(inplace=True)`. Checking for the shape of the dataframe using `df.shape()` returned (276, 16).

Now that the dataset is clean of all missing values and duplicate rows, it is ready for data exploration. However, slight preprocessing is required before moving on to data exploration. Since the GENDER column contains values “MALE” and “FEMALE”, it is necessary to encode these to comply with the other variables in the dataset such that “MALE” = 1 and “FEMALE” = 0 in the clean dataset. A similar

encoding is required for the LUNG\_CANCER column which contains values “YES” and “NO”. These values are encoded such that “YES” = 1 and “NO” = 0. The following is a snapshot of the clean dataset:

	GENDER	AGE	SMOKING	YELLOW_FINGERS	ANXIETY	PEER_PRESSURE	CHRONIC_DISEASE	FATIGUE	ALLERGY	WHEEZING	ALCOHOL_CONSUMPTION	COUGHING	SHORTNESS_OF_BREATH	SWALLOWING_DIFFICULTY	CHEST_PAIN	LUNG_CANCER
0	1	65	1	2	2	1	1	2	1	2	2	2	2	2	2	1
1	1	74	2	1	1	1	2	2	2	1	1	1	2	2	2	1
2	0	68	1	1	1	2	1	2	1	2	1	2	2	1	2	0
3	1	65	2	2	2	1	1	1	1	1	2	1	1	2	2	0
4	0	61	1	2	1	1	1	1	2	1	2	2	2	1	1	0

Figure 3: First 5 rows of the clean dataset

Additionally, since variable AGE is a continuous variable, it’s required to scale the column so that it is easier to train the ML models accurately without letting AGE have an unfair advantage over other categorical variables. In order to do that, we scale the variable AGE using scalar fit transformation:

```

scaler = StandardScaler()
X_train[AGE] = scaler.fit_transform(X_train[[AGE]])
X_test[AGE] = scaler.transform(X_test[[AGE]])
X_train.head()

```

	GENDER	AGE	SMOKING	YELLOW_FINGERS	ANXIETY	PEER_PRESSURE	CHRONIC_DISEASE	FATIGUE	ALLERGY	WHEEZING	ALCOHOL_CONSUMPTION	COUGHING	SHORTNESS_OF_BREATH	SWALLOWING_DIFFICULTY	CHEST_PAIN	LUNG_CANCER
221	0	0.891189	0	0	0	0	0	1	1	0	0	0	0	1	0	0
226	0	1.011464	1	0	0	0	0	1	1	0	0	0	0	1	0	0
234	0	0.138878	0	0	0	0	0	1	1	0	0	0	0	1	0	0
181	0	0.208222	1	1	0	1	1	0	0	0	0	0	0	0	0	0
214	0	-0.482078	0	0	1	0	0	1	0	0	0	0	0	1	0	0

Figure 4: First 5 rows of the dataset with scaled AGE variable

We store the mean and standard deviation of AGE from the clean dataset in two different variables before using scalar fit transformation on the entire column. This will be useful in the future to compute the scaled value of user input of AGE using the following formula:

```

[42] age_scaled(input("enter age"))
p=(age_age_mean)/(age_sd)
print(p)

```

enter age? 6.4881758855823475

Figure 5: Scalar fit transformation of user input of variable AGE

### C. Data Exploration

The clean dataset is now ready for data exploration. First, the dataset is divided into categorical and continuous variables. Since variable AGE is the only continuous variable, all other variables, including variable LUNG\_CANCER, are added to the categorical dataset. Once all the variables are separated into these two datasets, a visualization is created to understand the AGE variable:

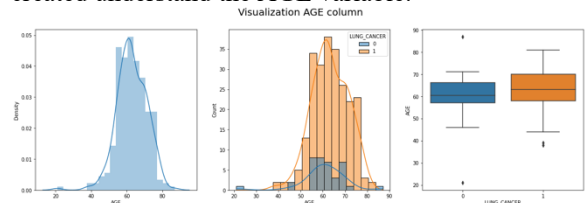


Figure 6: Visualizing AGE column

The following observations are taken from Figure 6:

1. The box plot on the right shows that there exist a few outliers in the dataset.

- Most observations fall within the range  $50 < \text{AGE} < 70$ .
- There exist more cases of “YES” to lung cancer than “NO” to lung cancer

Plotting a count plot for GENDER with LUNG\_CANCER as classification gives the following output:

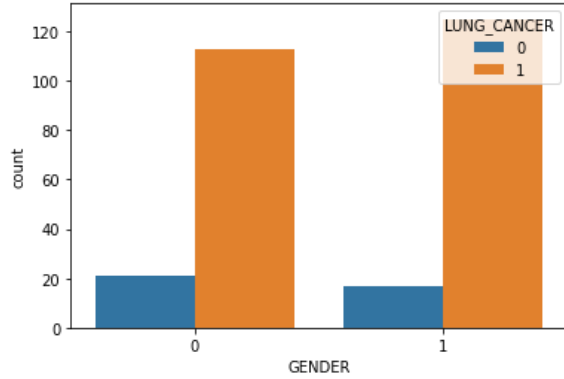


Figure 7: Countplot for GENDER with LUNG\_CANCER for classification

The following observations are taken from Figure 7:

- The count plot shows that the number of males that responded “YES” to lung cancer is more than the number of females with “YES” to lung cancer.
- The number of “YES” to lung cancer observations (male and female included) are much greater than the number of “NO” to lung cancer observations. There is a large discrepancy between the two classes in the LUNG\_CANCER column.

The last data exploration step was to understand the correlations between the variables using a heatmap:

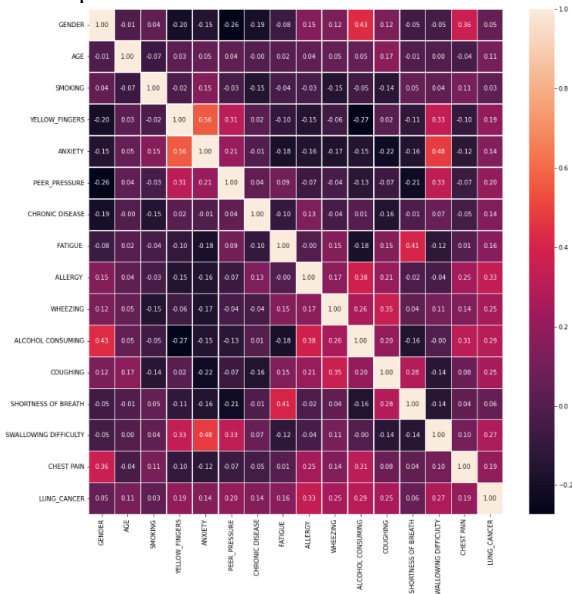


Figure 8: Heatmap indicating correlations between variables

The observation from Figure 8 is that there is a lot of multicollinearity between the variables in the dataset as indicated by the dark colored cells in the heatmap.

#### D. Training ML Models

It is now time to train several ML models to evaluate which one works best to predict lung cancer. The following are the ML models used and their scores:

##### i. KNeighborsClassifier

After training and testing the dataset on the KNeighborsClassifier model, the model achieved a precision of 1.00, sensitivity of 0.86, and F1-score of 93%. The following is a summary of the scores:

	precision	recall	f1-score	support
0	0.88	1.00	0.94	60
1	1.00	0.86	0.93	59
accuracy			0.93	119
macro avg	0.94	0.93	0.93	119
weighted avg	0.94	0.93	0.93	119

Figure 9: Result evaluation of KNeighborsClassifier

##### ii. Support Vector Machine

After training and testing the dataset on the Support Vector Machine model, the model achieved a precision of 1.00, sensitivity of 0.98, and F1-score of 99%. The following is a summary of the scores:

	precision	recall	f1-score	support
0	0.98	1.00	0.99	60
1	1.00	0.98	0.99	59
accuracy			0.99	119
macro avg	0.99	0.99	0.99	119
weighted avg	0.99	0.99	0.99	119

Figure 10: Result evaluation of Support Vector Machine

##### iii. LGBM Classifier

After training and testing the dataset on the LGBM Classifier model, the model achieved a precision of 1.00, sensitivity of 0.90, and F1-score of 95%. The following is a summary of the scores:

	precision	recall	f1-score	support
0	0.91	1.00	0.95	60
1	1.00	0.90	0.95	59
accuracy			0.95	119
macro avg	0.95	0.95	0.95	119
weighted avg	0.95	0.95	0.95	119

Figure 11: Result evaluation of LGBM Classifier

#### E. Choice of ML Model & Evaluation

After implementing all 3 ML models to the training and test dataset, the results of each of the

models were examined and scrutinized. The following table summarizes the results:

ML MODEL	MEASURE	VALUE
KNeighborsClassifier	Precision	1.00
	Sensitivity	0.86
	F1-score	0.93
<b>Support Vector Machine</b>	<b>Precision</b>	<b>1.00</b>
	<b>Sensitivity</b>	<b>0.98</b>
	<b>F1-score</b>	<b>0.99</b>
LGBM Classifier	Precision	1.00
	Sensitivity	0.90
	F1-score	0.95

Table 1: Summary results of all three ML models

It is clear from Figure X that the Support Vector Machine model performs best with a sensitivity score of **0.98**. It is the measure of sensitivity that indicates the number of true positives that the model predicts based on the train and test dataset. This measure is the most important measure in evaluating ML models for healthcare dataset.

### III. APPLICATION

A total of three ML models were implemented and evaluated using the clean dataset in the previous section. From the three models, the Support Vector Machine model showed the best results with a sensitivity score of **0.98**. Therefore, this model was used to create the Lung Cancer Detection application.

The application would take 15 user inputs for all 15 attributes defined in the dataset in 0s (for “NO”) and 1s (for “YES”). These 15 user inputs will then be stored in a NumPy array. This array is then converted into a Python list that is then fed into the trained SVM model to predict whether or not the user is at a risk of lung cancer.

Since we scaled the AGE variable (check II. Methodology, Data Cleaning and Preprocessing), it is necessary to scale the user input of AGE before feeding all inputs into the trained ML model. This is done using the `age_mean` and `age_sd` of AGE from the clean dataset:

```
[4]: age_scaled = (age - age_mean) / age_sd
print(age_scaled)

after age:
0.4027380811475
```

Figure 12: Scaling user input of AGE using scalar fit transformation

The following is a snapshot of the user inputs and the prediction made by the trained SVM model:

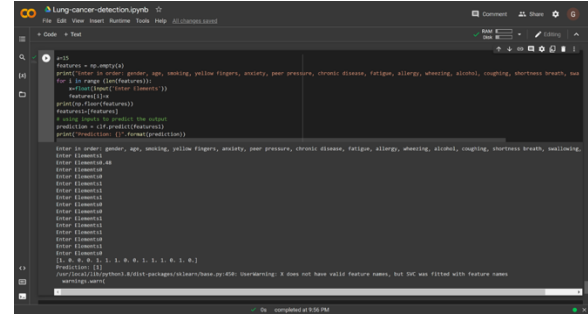


Figure 13: Snapshot of Lung Cancer Detection application (1 = “YES” for lung cancer)

Since the ambition behind this project was to incorporate the skills learned in this course and use big data tools, the ML application developed in this project was then implemented and run on Microsoft Azure Automation. MS Azure Automation is a cloud-based automation and configuration service that provides a user-friendly Python environment to implement and run Python programs [6]. The results obtained on local machines were successfully replicated in the MS Azure Automation environment.

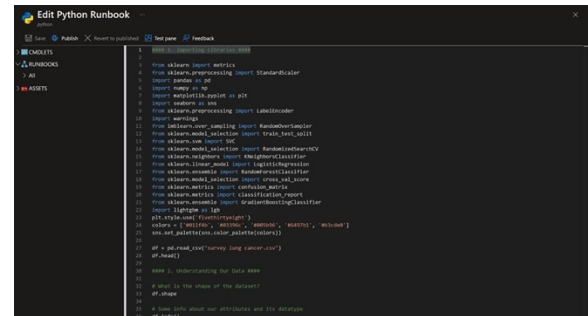


Figure 14: Running the application on MS Azure Automation

### IV. FUTURE WORKS

There exists many other ML models that might yield better results than the Support Vector Machine model used in this project. Future direction for this project would be to test other ML models to get better results for lung cancer prediction. Additionally, obtaining a bigger dataset with more observations would also help train existing and future ML models to predict more accurately. Moving forward, another ambitious approach for this project

would be to create a functional web-based ML application hosted in a cloud environment for public use.

## V. CONCLUSION

Lung cancer detection in its early stages is crucial for minimizing health risks among at-risk individuals. This project was intended to first train, test, and evaluate the best ML model predicting lung cancer from a dataset. The Support Vector Machine model returned the best outputs with a precision of 1.00, sensitivity of 0.98, and F1-score of 99%. This model was then implemented to create a ML application that can predict the risk of lung cancer based on user input for 15 questions. Finally, a functional application was created using the SVM model and implemented into Microsoft Azure Automation to predict lung cancer.

## REFERENCES

- [1] MYSAR AHMAD BHAT. "Lung Cancer." *Www.kaggle.com*, [www.kaggle.com/datasets/mysarahmadbhat/lung-cancer](http://www.kaggle.com/datasets/mysarahmadbhat/lung-cancer).
- [2] "Sklearn.preprocessing.StandardScaler — Scikit-Learn 0.21.2 Documentation." *Scikit-Learn.org*, 2019, [scikit-learn.org/stable/modules/generated/sklearn.preprocessing.StandardScaler.html](http://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.StandardScaler.html).
- [3] Joby, Amal. "What Is K-Nearest Neighbor? An ML Algorithm to Classify Data." *Learn.g2.com*, 19 July 2021, [learn.g2.com/k-nearest-neighbor](http://learn.g2.com/k-nearest-neighbor).
- [4] scikit learn. "1.4. Support Vector Machines — Scikit-Learn 0.20.3 Documentation." *Scikit-Learn.org*, 2018, [scikit-learn.org/stable/modules/svm.html](http://scikit-learn.org/stable/modules/svm.html).
- [5] "Lung Cancer Facts." *Lung Cancer Research Foundation*, [www.lungcancerresearchfoundation.org/lung-cancer-facts/](http://www.lungcancerresearchfoundation.org/lung-cancer-facts/).
- [6] SnehaSudhirG. "Azure Automation Overview." *Learn.microsoft.com*, [learn.microsoft.com/en-us/azure/automation/overview](https://learn.microsoft.com/en-us/azure/automation/overview).