Data Science 316 A1 Project

Project Team:

Arlo Steyn
24713848

Andre van der Merwe
24923273

Stellenbosch University

Date of Submission: 18 April 2024

Contents

1 Problem Statement

1.1 Introduction to the Lung Cancer Prediction Approach

The field of healthcare and human welfare has been one of the most important fields of science since the beginning of mankind. It is the very reason that our species is still alive and thriving today. Unfortunately, many people do not have access to the same healthcare resources in the world, particularly in third-world countries such as South Africa.
For this very reason, we have decided to try and reduce this gap by helping countries that are less fortunate. Many people in countries such as South Africa do not have access to or enough money to seek medical treatment of any kind.
In a country where many people work in mines and have terrible smoking habits, it is often the case that people develop lung cancer and have no idea as well as a lack of funding to investigate the issue further. This is where our lung cancer classification model comes into play. With only a simple questionnaire, the data gathered can help us point a worried civilian in the right direction by classifying them into one of 3 categories:

1. Low Chance of having lung cancer,
2. Medium Chance of having lung cancer,
3. High Chance of having lung cancer.

This initiative aims to bridge the healthcare gap, offering crucial early guidance to those potentially facing lung cancer in underprivileged regions.

1.2 Problem Description

The intention of this classifier is to help mitigate the monetary and accessibility barrier for those who otherwise would not have access to these resources, thus improving the state of welfare in underprivileged countries. The idea is to collect data from patients in a cost-effective and accessible manner with a somewhat simple questionnaire, whereby a qualified individual will gather information (needed for the classifier) about patients. This data will then be used (with the help of the classification model) to predict whether or not a patient should take further steps in their journey with lung cancer. The goal is to be as accurate as possible in our predictions, as it can potentially be the difference between life and death.

However, there are challenges that we may encounter, including:

- **Human Error:** Our qualified personnel could possibly enter data incorrectly, leading to incorrect predictions and referrals.
- **Access to Treatment:** A potential high-risk patient may not be able to receive the treatment they need or pursue further investigations.
- **Accuracy Concerns:** False positives and false negatives can be critical, impacting life or death decisions.
- **Data Collection:** It may be challenging to collect data in rural areas, limiting accessibility.

- **Representation and Bias:** The data may not be representative of the entire population, resulting in models that are less effective for certain groups of participants and providing incorrect results.
- **Stage of Lung Cancer:** Due to biases in the underlying data, our model may struggle to accurately differentiate between the early and late stages of lung cancer.

Our commitment is to overcoming these challenges to enhance the effectiveness and reliability of our model, aiming for the best possible outcomes in improving healthcare accessibility and accuracy in underprivileged regions.

1.3 Objectives

Our objective is to develop a classification model that can accurately categorize participants into the specified groups. Given the critical implications of misclassification, where false negatives could lead to a scenario where a patient in dire need of treatment is overlooked, and false positives could result in unnecessary stress, financial burden, and further medical examinations for those not at ris, we prioritize the reduction of both false positive and false negative rates. Ultimately, our goal is to enhance the predictive accuracy of our model, ensuring it serves as a reliable tool in medical diagnostics.

The following is a broad overview of the steps we will follow to achieve the above:

1. **Exploratory Data Analysis (EDA) and Data Preparation:** Initial analysis to understand the dataset's characteristics followed by cleaning and preprocessing to ensure data quality and readiness for modeling.
2. **Model Development and Evaluation:** Training various predictive models to identify the most effective approach, based on their performance metrics.
3. **Mitigation of Overfitting:** Implementing strategies to prevent overfitting, ensuring our models generalize well to new data.
4. **Comparative Analysis:** Synthesizing our findings and comparing them with the outcomes derived from the *current approach* to highlight advancements or insights gained.
5. **Validation and Refinement:** Further validating the selected model(s) against unseen data and refining the approach based on feedback and performance evaluations.
6. **Evaluation of Findings:** Preparing a comprehensive report detailing our methodologies, results, and comparative analysis to contribute to the broader understanding of lung cancer prediction.

1.4 What has already been done in the field?

[1]Lung cancer ranks as the leading type of cancer on a global scale and in 2018 alone, lung cancer accounted for more than 2.1 million known new cases as well as 1.8 million known deaths.
[2] $\frac{2}{3}$'s of the time, patients are diagnosed with lung cancer at a later stage where their

chances of survival are between 4% and 28%, compared to a 55% survival rate if founded at the early stages.
More specifically, in South Africa, lung cancer is part of the top 3 cancers in men and among the top 7 in woman but, is the leading cause for cancer deaths.
In South Africa, awareness surrounding lung cancer remains significantly limited. Given the crucial role of early detection in improving prognosis, it's important for individuals to become adept at identifying the early indicators of lung cancer. This claim is emphasized by the findings of our project, which highlights the necessity for increased educational initiatives aimed at sharing knowledge about the symptoms and warning signs of lung cancer to the South African population.

1.5 Overview of the *Current Approach*

The *current approach,* as detailed by **ANAGHA K P** in the study [3] "Lung Cancer Prediction using Logistic Regression Model," was sourced from [4] Kaggle and published towards the end of 2023. The *current approach* effectively performs statistical modeling on a dataset of patients who were tested for lung cancer, and clasifies them into one of the three above classes.
In the following sections, this approach will be explored in greater detail.

2 Data Description

2.1 Data Source

The dataset of the *current approach* was also found on [4] Kaggle, by the name [5] "Lung Cancer Prediction". According to the dataset page on Kaggle [5], data was collected from a sample of people in China where participants were followed for an average of six years. Participants were also divided into two groups: participants who lived in a high levels of air pollution and participants who lived in areas with low levels of air pollution.
[6] The studies aim was to find if air pollution was linked to an increased chance of having lung cancer, even for non-smokers. Note that the outcome of the study does not imply that a higher level of air pollution increases your chances of getting lung cancer, it implies that there may be a link between the two and that higher air pollution can play a role in ones chances of getting lung cancer.

A 3rd world country, like South Africa, can benefit from this study due to the contrast in rural and non-rural areas. Like China, South Africa also has many cities and other areas that contains high levels of air pollution as well as areas that do not. Similarly, (Like most 3rd world countries) South Africa also has a dense population of smokers.

2.2 Dataset Characteristics

The dataset, titled *"cancer patient data sets.csv"*, comprises data derived from the studies discussed above. It includes 1,000 unique entries (with no repeated patient records), each characterized by 26 distinct features. These entries are systematically categorized based on a "Level" attribute, which assesses the likelihood of an individual being associated with lung cancer. This categorization is executed through three ordinal labels:

- **1:** Low risk of lung cancer,
- **2:** Medium risk of lung cancer,
- **3:** High risk of lung cancer.

In summary, the dataset provides a comprehensive analysis framework for lung cancer risk assessment, encapsulated by the following key metrics:

- **N**: The total number of unique patient entries, amounting to 1,000.
- **Features**: Each entry is described through 26 distinct features, offering a multi-dimensional perspective on patient data.
- **P**: The dataset categorizes lung cancer risk into 3 ordinal levels, signifying low, medium, and high risk.

This structured dataset facilitates detailed statistical analysis and predictive modeling in lung cancer research.

Class Imbalances:
The distribution of the *Level* response variable within the dataset reveals a noteworthy composition: 303 instances of Class 1, 332 instances of Class 2, and 365 instances of Class 3. This distribution indicates a relatively balanced dataset, albeit with a slight predominance of Class 3. The balanced distribution of risk level; low, medium, and high, within the dataset is particularly advantageous for lung cancer risk assessment studies. It ensures that the (future) predictive models developed are well-trained across all risk categories, crucial for accurately identifying the influence of air pollution on lung cancer risk. This balanced representation aids in mitigating the risk of model bias towards more frequently occurring classes and enhances the model's ability to generalize across the spectrum of lung cancer risk levels, facilitating a more precise and equitable assessment of air pollution's role as a risk factor.

2.3 Potential Data Problems

The quality of our data plays a crucial role in the reliability and analysis of our findings. It is thus critical that we identify and understand potential problems in the data itself in order to ensure the accuracy of our analysis. Potential data problems in our dataset include:

- **Handling of Missing and Null Values**: The data is fully intact with there being no missing values as well as no null values. This simpifies the data cleaning process.
- **Categorical Target Variable:** Our target variable variable, *Level*, is categorical and needs to be converted to a numeric value. Dummy variables will be used for this and added in the place of the current target variable variable. This allows our model to interpret these values and make predictions.
- **Redundant Features:** The first two features of our dataset, *index* and *Patient Id* are redundant in nature as their sole purpose is to keep track of numeration as well as identification of patients in a database respectively. These features will be dropped as to reduce the noise as well as the dimensionality of our future model(s).
- **Dominance of Categorical Variables:** 23 of the 24 remaining features are categorized as ordinal numbers, spanning from 1 to 10. While the structured categorization helps us to distinguish between different levels of variability, it presents the following challenges:

    1. **Ordinal Nature:** The ordinal scale implies a order or hierarchy but does not quantify the magnitude of difference between all categories. This can complicate the modeling process, as standard numerical operations may not accurately reflect the relationships between features.
    2. **Assumption of Equidistance:** Treating these categories as numerical values assumes equal spacing between them, an assumption that may not hold true and be misleading in analysis and model generation, leading to biased outcomes.
    3. **The Remaining Non-Ordinal Feature:** *Age*, our only non-ordinal feature captures the age of a patient. The age of a patient can range anywhere from 0 - 120, introducing a large scale discrepancy compared to the other narrowly ranged categorical variables. Such a discrepancy may pose challenges

in training due to the higher variance in *Age* which will disproportionately influence the future model(s) predictive power. To prevent this, normalization techniques will be used to appropriately scale this feature.

It is of utmost importance to take these problems into account in order for our potential model(s) to have the best possible outcomes for prospective patients.

3 The *Current Approach*

This section serves as a comprehensive overview of the [3] *current approach*.

3.1 Exploratory Data Analysis

The author starts off by cleaning the data. This is done by performing the operation "*df.isnull().sum()*", which sums up all null values per feature. Fortunately, there are no null or missing vlaues in the data, meaning that imputation methods were not required. This is significant as the integrity of the underlying data is preserved which leads to more accurate results in the long run.
Our author then, explores the correlation heatmap, allowing for a visual representation of correlation amongst features. This is followed by another heatmap wherby features are contrasted by correlation in relation to the target variable *Level* and sorted in descending order. The respective heatmaps are as follows:

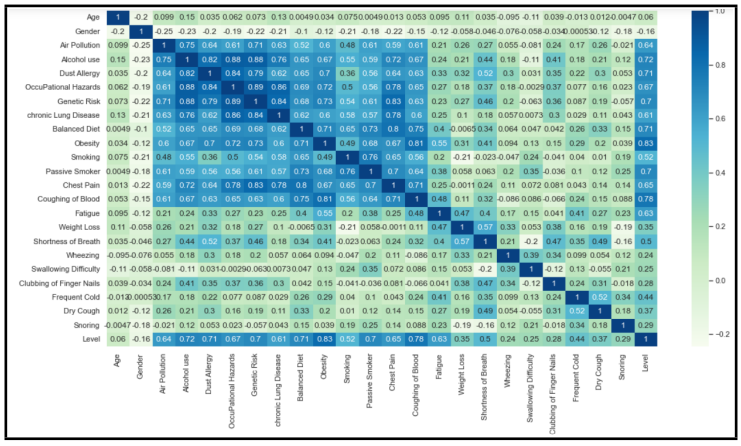


Figure 1: Heatmap for the Correlation amongst all features and target variable

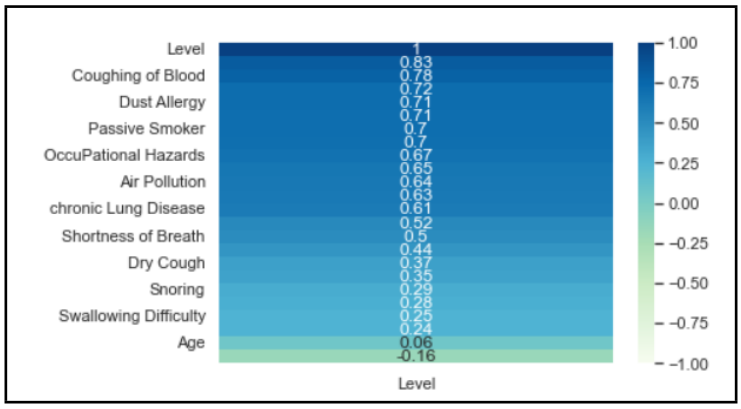


Figure 2: Heatmap of the features correlating to *Level*

3.2 Pre-Processing Steps

Fortunately there was no missing data in the data set. So the Author did not have to deal and find solutions to missing data in the data set.

Our author drops the *Index* and *Patient Id* columns due to them being redundant in terms of the predicitive information they contain for classification. The categorical(ordinal) target variable, *Level*, is replaced with numerical values (such as 1, 2 and 3 respectively) in order for our model to interpret these values and make predictions.
The Author did not check if there are any outliers, that might greatly effect the training of the model.
Lastly the dataset is split into X(predictors, n=10) and y variables(target variable: *Level*) to prepare for training and testing of the model. The variables are split into training and test sets respectively by using the "*train_test_split*" function. The following parameters were used for the function:

- **Test Size:** Allocates 30% of the data for testing and 70% of the data for training the model.
- **Random State:** Randomly decides which datapoints will be used for the respective training and test sets. A value of 1 is used, meaning that the same data, more or less, will be used for the same training and test splits.

Again, we are unsure how the author decided on these paramaters. It is evident that no cross validation or any other such methods were used to test which training and test sizes are best. The author did not go into depth for optimal training measures.

3.3 Models Considered

Logistic Regression is the only model used for the *current approach*.
Figure 2 above in section 3.1 shows the correlation between *Level* and each other variable in a descending order and on the left hand side it displays some of the variables names, such as *Age* and *Snoring*. Not all of the feature names are present here. The Author decided to choose the 11 features on the heatmap of the features correlating to *Level* as the subset of variables that the future model will be trained on. This effectively means that the Author just randomly chose 11 variables without performing any form of variable selection or adding a regularization term. This can be problematic as it can lead to models that may not capture the full complexity of the underlying relationships of the data. Its also concerning as it misses key insights from features with less non-linear relationships to the target variable.

It is evident that the author possibly lacks insight or simply just glanced over this step due to the minimal amount of effort put in as well as the lack of understanding displayed. This gives us great room for improvement in the *Improved Approach* coming in later sections.
The authors model is built with only 1 parameter: *Solver = liblinear*. This specifies the algorithm used for optimisation in solving the logistic regression problem. It is used to handle linearly seperable data and is useful for quick and scalable results in a high dimensional dataset. *LibLinear* is best used for binary classification tasks, but we are working with a multi-class classification problem. Although *liblinear* can still classify binary-class problems with decent outcomes, it is not ideal.
The author does not use any other tuning parameters, and for the parameter that he/she does use, it is not completely correctly implemented.

3.4 Performance Measures Considered

Using the Logistic Regression classifier, predictions are made on the test data. These predictions are then evaluated by comparing the predicted values $\hat{y}_{\text{test}}$ with the true values y_{test}. To assess the performance, *SKLearn's Accuracy Score* module is used, which reveals an accuracy of 90.3%. This result is notably high, considering the simplicity of the preprocessing and model selection approach.
To explore these results further, a confusion matrix is generated. A confusion matrix provides detailed insight into the accuracy of a classification model by showing the true positives, true negatives, false positives, and false negatives etc., enabling a deeper analysis of the model's performance beyond overall accuracy.
The following displays the discussed confusion matrix for the *current approach*.

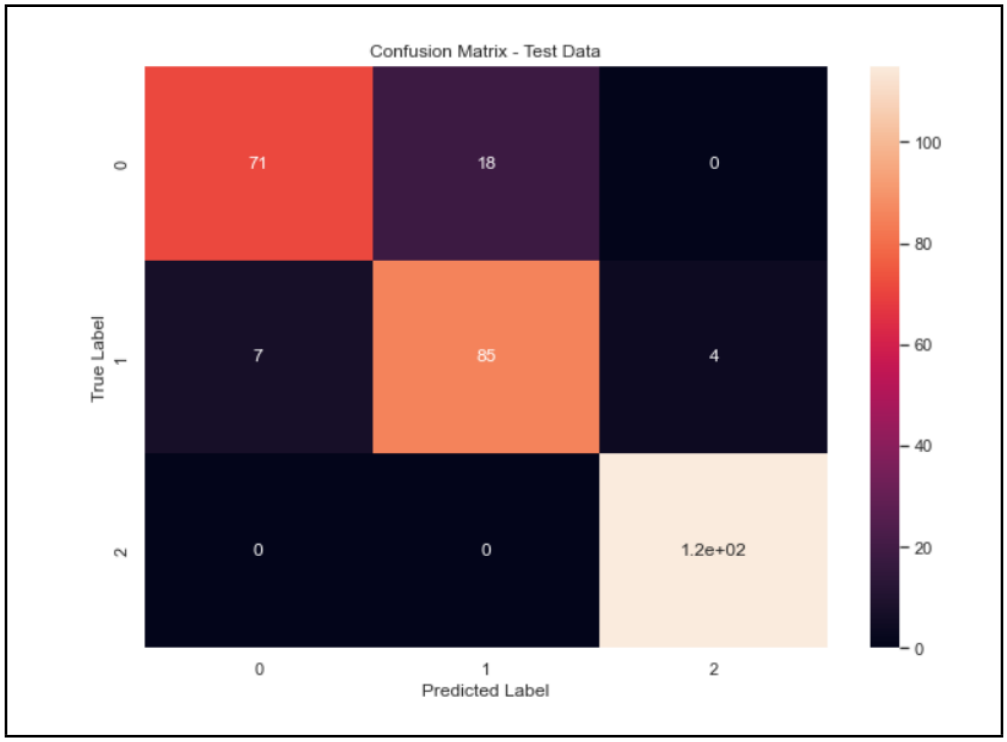


Figure 3: Confusion Matrix

This model clearly struggles with the following:

- **False Negatives for label 0:** The model incorrectly predicted 18 instances of label 0 as label 1, potentially delaying treatment for patients with possible lung cancer.
- **False Negatives for label 1:** There were 7 instances misclassified as label 0 and 4 as label 2, meaning some medium-risk patients might not receive the possible timely interventions they need.

What the model did well:

- **True Positives for label 0:** The model correctly identified 71 instances of label 0, which is crucial for patients with no or possible lung cancer, ensuring they are not subjected to unnecessary treatments.

- **True Positives for label 1:** Correctly predicting 85 instances of label 1 allows for accurate identification of medium risk patients, facilitating appropriate monitoring and potential intervention.
- **True Positives for label 2:** With 115 correct predictions for label 2, the model effectively identifies high risk patients, who can then be prioritized for aggressive treatment and management, which is essential for improving their prognosis.

The author concludes their notebook with a classification report that compares the predicted labels ($\hat{y}_{\text{test}}$) against the true labels (y_{test}). The following is the report used:

	precision	recall	f1-score	support
1	0.91	0.80	0.85	89
2	0.83	0.89	0.85	96
3	0.97	1.00	0.98	115
accuracy			0.90	300
macro avg	0.90	0.89	0.90	300
weighted avg	0.90	0.90	0.90	300

Figure 4: Classification Report

The classification report provides crucial metrics that inform the effectiveness of the predictive model [7]:

- **Precision:** High precision indicates the model's strong capability to correctly identify patients with possible lung cancer, ensuring those without the disease are not subjected to unnecessary medical procedures.
- **Recall:** The recall metric suggests how well the model can capture all relevant cases of possible lung cancer. Lower recall could mean some patients with the disease might not be identified, which is critical for conditions that require early detection.
- **F1-Score:** This score balances precision and recall and is particularly useful when the costs of false positives and false negatives are roughly equivalent. A high f1-score is indicative of a model's robustness in classification tasks.

Many improvements can and will be made by our *improved approach* in subsequent sections. There is large room for improvement.

4 Potential Problems with the *Current Approach*

The following will be based off various sections of Section 3.

4.1 Problems with the EDA Step

The *Current Approach*, Section 3: 3.1

Summary of this Step:
The author has a minimalistic approach to EDA. Operations are done to see if there are null or missing values in the dataset, which there are non of. A correlation heatmap is then printed followed by a heatmap of features correlating to the target variable Level.

Potential Problems and Shortcomings:
Due to the authors minimalistic approach, the following possible shortcomings are outlined:

- **Insufficient Data Understanding:** Due to the lack of EDA done, the author misses out on possible potential insights that could've been gained such as anomalies and data distributions. The author seems to not take note of the ordinal nature of the data and how it can influence his/her models and outcomes in the future. Furthermore the author also seems to lack understanding of where data lies such as the skewness and kurtosis of the data.

- **Class Imbalances:** The author does not verify class imbalances. Luckily for the author, the classes are balanced. If this was not the case, the author would struggle for accurate predictions to be made for certain classes as there would be too little or too much training data for a specific class.

- **Outliers and Duplicate Values:** The author fails to check for outliers. This is a crucial step in EDA as these datapoints can distort the statistical distributions of variables and alter the performance of your model due to models being potentially sensitive to extreme value. It also makes it difficult to interpret trends or patterns in the data. Duplicate values are also overlooked, this is problematic as this can lead to a model with increased bias due to the over representation of the same data point.

Extending, Potential Benefits & Critical Insight:
To improve upon the authors shortcomings on the EDA Step, the following is to be considered:
In depth EDA techniques need to be implemented such as box-plots, scatter plots and histograms. This ensures a better understanding of the data one is working with and also allows one to identify outliers in the data. Depending on the severity of the outliers in the dataset, use a subset of the dataset that minimizes outliers or just completely remove them from the dataset. This not only enhances data visualization and understanding of the underlying patterns of the dataset, but also could lead to a better model in the future due to less outliers skewing the distributions on the data. In terms of duplication, an approach for removing duplicate values is as follows: *Patient Id* represents each patients entry. Removing rows with the same id can be beneficial to us because our model will

have less bias. It is important to note that a patient can possibly have 2 or more entries depending on the time interval of tests.

4.2 Problems with the Pre-Processing Step

The *Current Approach*, Section 3: 3.2

Summary of this Step:
The author cleans the data by dropping 2 redundant features (*Index* & *Patient Id*). The categorical (ordinal) target variable, *Level*, is then transformed into numercial values (1, 2, 3) so that the data can be interpreted by the statistical model. The author also checks if there are any null values or missing data, which there is non-of. The author splits the new cleaned and processed dataset into X (predictors: 10 features) and y (target variable: *Level*) variables and then splits the these into training(70% of observations) and test(30% of observations) sets in preperation for training and testing the model. The data partitioned will be almost the same each time when running tests as the author chooses a random state of only 1 when splitting.

Potential Problems and Shortcomings:
The following are highlighted:

- **Random State of 1:** As discussed earlier, the random state parameter controls the variation of datapoints chosen from the dataset for the partitioning of train and test sets. Only chosing a value of 1 means that there is little to no variation, meaning that the subsets are more or less the same set each time. This is problematic because, the robustness of the model is not tested against various random splits. This means that the model will generalise well to the data but will not perform as well on unseen training and test data, leading to sub-optimal results.
- **Train and Test Set Split:** Allocating 70% of the data to the train set may cause the model to learn noise and increase the likelihood of overfitting. Similarly, allocating 30% of the data to the test set may not fully encompass the models performance on various scenarios. The point being that the author seemingly did no tests to test for optimal train and test splits.

Extending, Potential Benefits & Critical Insight:

- **Dynamic Random State Selection:** One can employ a selection of random state selections per model generation. This enhances the models robustness and stability on diverse data subsets. This aids in mitigating overfitting which produces a more reliable model.
- **Cross Validation:** Instead of having set values for train and test splits, one could use k-fold cross validation. This ensures optimal data partitioning which also aids in the robustness of a model as discussed previously.

4.3 Problems with the Models Considered Step

The *Current Approach*, Section 3: 3.3

Summary of this Step:
In terms of pre-processing and dimentionality reduction, it seems that the author randomly selected 10 features alongside the target variable, *Level*, out of the remaining 24 variables for his/her new dataset. The author only uses a notably simplistic logistic regression model. Only one parameter is present in the model: *Solver = Liblinear* which optimises the logistic regression problem, particularly for binary classification problems. No tuning of paramaters or regularization techniques are present.

Potential Problems and Shortcomings:
Randomly selecting features to train a statistical model, as the author did, is incredibly problematic due to the following:

- **Overfitting:** Randomly selected features may include irrelevant information that does not contribute to the models ability to learn patterns in the underlying dataset. This leads to a model that generalizes poorly on unseen data.
- **Complexity & Inefficiency:** Randomly selecting variables results in many features that are redundant. This hinders the performance of the model as unnecessary variables will increase the complexity and hence, the computation time of the model.
- **Accuracy & Validation:** Due to the random selection of features, it is likely that a model will produce higher variance of performance across datasets. This makes results unreliable. Essential features will also be left out which can degrade a models predictive power.

Training a logistic regression model with only one parameter, and that paramater not being used optimally, can cause the following problems:

- **Complexity & Performance:** By only using one parameter, one is left with a very basic model which is followed by an increase of bias. It is also common practice to tune hyperparameters to maximize the performance and complexity trade-off of a model. This is simply not present and is problematic due to the subsequent decrease in model performance.
- **Regularization:** There are no regularization methods present. Regularization methods are crucial to ensure that models are not overfit, especially if a model is working in high dimensional space.
- **Multi-Class Problem:** As mentioned before, using a binary solving classification parameter for a multi-class problem is not ideal.

Extending, Potential Benefits & Critical Insight:
The follwoing can be done to greatly improve the authors approach:

- **Feature Selection:** The implementation of systematic feature selection methods such as backward selection can be beneficial as it ensures that only relevant features are used to train the model. By properly implementing feature selection, it not only reduces the computational costs of the model, but also leads to an increase in predictive power on unseen cases.

- **Hyperparameters & Regularization:** Tuning techniques such as *grid-search-cv* should be incorporated alongside regularization techniques such as L1 and L2 regularization. This ensures optimal results and also ensures that the right balance of bias and variance is found, which is critical for the performance on a model as it ensures optimal results for a models predictive power on unseen data.

- **Multi-Class Problem:** replacing the linear solving parameter for the logistic regression model to a multi-class solver such as *'newton-cg', 'sag', or 'saga'* can prove to be beneficial. This is because these solvers can enhance a models ability to classify multi-class datasets which leads to better performance.

4.4 Problems with the Performance Measures Considered Step

The *Current Approach*, Section 3: 3.4

Summary of this Step:
The author uses the trained model and makes predictions for the test set. The author then compares the results of the outcomes of the test sets' actual values, yielding an accuracy of 90.3%. A confusion matrix is then utilized to delve deeper into the results and analysis of the models results on the new dataset.

Potential Problems and Shortcomings:
The author only checks the accuracy of the model and generates a confusion matrix. This has a number of problems:

- **Lack of Depth:** The author does generate a confusion matrix and also gets the accuracy of the model but fails to go into detail about what the results entail. He/She fails to relate the results to the core idea of the problem at hand.

- **Singular Model Focus:** Without considering other models, the author missed out on opportunities to possibly improve upon his model and possibly get better results.

- **Lack of Model Validation Techniques:** The author does not test his/her model on different subsets of data and in combination to the training test sets' random state being only set to 1, the reliability and stability of the models predictions will certainly vary due to no cross-validation or bootstrapping methods used.

- **Model Assumptions:** The author fails to mention or interpret the assumptions of his/her logistic regression model. In the case where the predictions that a model makes could alter the lives of patients, it is imperative to be as accurate as possible. By not verifying the assumptions made on the data, such as independence of observations or multicollinearity among predictors, can lead to unreliable and incorrect results.

Extending, Potential Benefits & Critical Insight:
To address the main concerns of the authors implementation of the performance measures considered step, the following can be done to extend the approach:

- **Utilization of Broader Performance Merics:** We would implement more evaluation techniques such as ROC-AUC curves as this would give a greater reflection of the models performance and provide greater insight to potential problems with the models results. Furthermore, we would dive deeper in the analysis of the confusion matrixs' results by discussing the recall, precision and f1-score generated and its impact on the problem at hand. In Figure 3 and 4 of section 3.4, this analysis was already done.

- **Cross Validation Implementation:** K-fold cross validation would be applied. This would ensure that the model does not generalise to the data too well and also ensures that model is effective accross multiple subsets of data which mitigates sampling bias. This allows us to be more confident about the predictive power of our model on unseen cases.

- **Comparing Model Performance:** With our newly added models (which will come in the following sections), we would compare their results as well as the bias-variance trade-off for each and then also look at the complexity of each model to determine the most optimal model for our final selecton. This would maximise our chances in having the best possible predicitve power.

5 The New Approach

5.1 Exploratory Data Analysis

Before we can work and build models on the data, we have to do a thorough Exploratory data analysis to analyze and investigate the dataset and summarize their main characteristic.
We start our Exploratory data analysis off by searching for any null values in the data set. Fortunately there wasn't any missing values. We then search for any duplicates and outliers in our dataset and after a thorough investigation, we couldn't find any outliers or duplicates. There after we had to see if there were any categorical variables in the data set. This is a very important step, because any categorical data needs to be transformed into a dummy variable so that we can use this categorical data when we build our models. We also dropped the "index" and "Patient Id" features as they have no effect and provide no value when building our models.
Lastly we need to investigate the relationship between the features and for the sake of our classification model, we investigated the relationship between our target variable against each feature. We obtained the following boxplots:

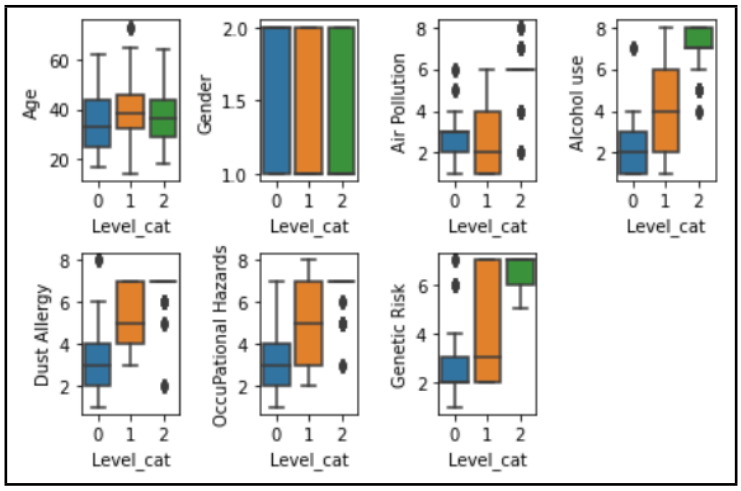


Figure 5: Boxplots of each level category against the features: Age, Gender, "Air Pollution", "Alcohol use", "Dust Allergy", "Occupational Hazards", "Genetic Risk"

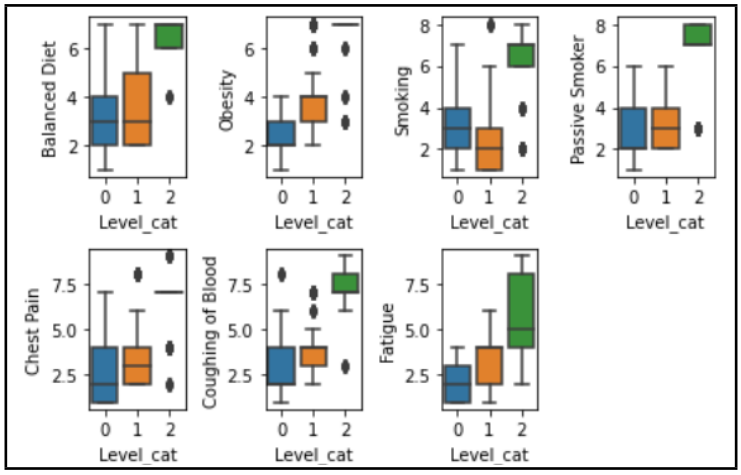


Figure 6: Boxplots of each level category against the features: "Balanced Diet", Obesity, Smoking, "Passive Smoker", "Chest Pain", "Coughing of Blood", "Fatigue"

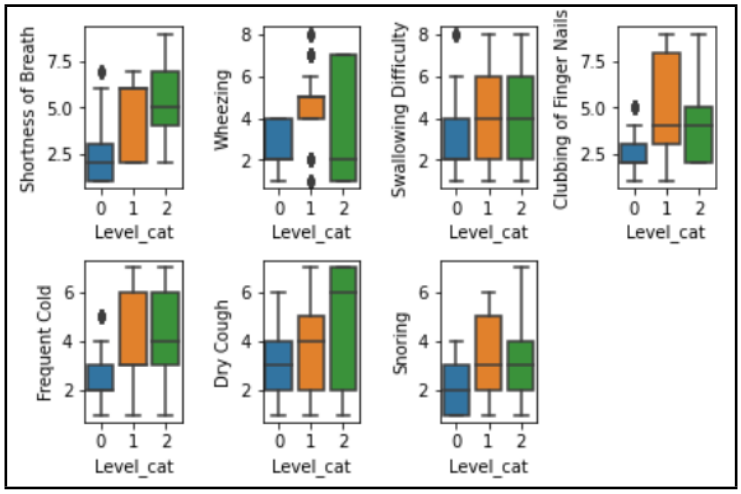


Figure 7: Boxplots of each level category against the features: "Shortness of breath", Wheezing, "Swallowing Difficulty", "Clubbing of Finger Nails", "Frequent Cold", "Dry Cough", Snoring

By investigating all of these boxplots, we can get a better understanding of which features have a much greater impact on our target feature. A lot of outliers in the boxplot indicates that the feature doesn't have a great relationship with our target feature. Also if the range from the first quantile to the third quantile is quite big, it means that there is a lot of variation between most of the data, so we want to have a smaller range. Knowing this, we can look for the features that has a good relationship with our target variable.
From our Boxplots, we gained the following features that doesn't have a good relationship with our target variable:

- "Gender"
- "Air Pollution"
- "Dust Allergy"
- "Occupational Hazards"

- "Chest pain"
- "Obesity"
- "Wheezing"

Features we found too have quite a good relationship with our target feature were:

- "Balanced Diet"
- "Passive Smoker"
- "Fatigue"
- "Swallowing Difficulty"
- "Frequent Cold"
- "Snoring"

5.2 Pre-Processing Steps

We have all of our information on the data, so now we have to use this information and set pre process our data, so that we can use the data to build our models and make predictions.
Fortunately there were no missing values that we had to deal with, as well as any outliers or duplicates that we needed to handle. Then as mentioned in the Exploratory data analysis, we added dummy variables to change the categorical data into data that we can use to build our models. The only feature we had to change was the target feature "Level", that described a persons chance of having lung cancer. After this we dropped the "index" and "Patient Id" features from the data.
When we constructed and tested our models, we never made use of a train test split. instead we opted for cross validation in order to get multiple results from our model, in order to optimize the build of our model. We used a number of K-fold cross validations to build our model with various different sized training sets and then test these models on various sized test sets. We opted to test them all with a K-fold cross-validation where:

- K was equal to 5
- K was equal to 10
- K was equal to 20
- K was equal to 50

We also didn't do any variable selection, dimensionality reduction or regularization in our pre-processing steps, as we decided to execute these steps when we worked on our model.

5.3 Logistic Regression Model(s)

We investigated three types of Logistic regression models[8], to determine the best possible logistic regression model that we can build in order to make correct and precise predictions on the dataset. These three types of logistic regression models were:

- Logistic regression using Lasso(l1) regularization
- Logistic regression using Ridge(l2) regularization
- Logistic regression using both Lasso and Ridge(elasticnet) regularization

Why we investigate Lasso regularization
The objective function gains a penalty term from Lasso regression that is equal to the absolute value of the coefficients. As a result, the model becomes sparse, with many of the coefficients having precise values of zero. For feature selection, Lasso regularization is helpful since it can automatically locate and eliminate features from the model that are unnecessary or redundant.

Why we investigate Ridge regularization
This increases the objective function by a penalty term that is equal to the square of the coefficients. As a result, the model has all of its coefficients almost but not completely equal to zero. Compared to L1 regularization, L2 regularization is less prone to overfitting.

Why we investigate Elastic Net regularization
By including a penalty component within the objective function that combines the absolute value and square of the coefficients, Elastic Net combines L1 and L2 regularization. As a result, the model has some coefficients that are close to zero and some that are equal to zero. When there are linked features in the data and Lasso tends to choose just one of them, Elastic Net can be helpful.

Benefits of using these regularization terms
Regularization makes the model's coefficients (weights) small by introducing a penalty term to the objective function. As a result, the model becomes less complex and less prone to overfitting. This would greatly aid in the construction of the logistic regression model by lowering the number of non-zero coefficients and streamlining the process of choosing only one.

How we chose the regularization strength (hyperparameter)
In order to fit our logistic regression models with their respective regularization term, we will first have to determine an optimal value for our regularization strength.
An optimal value for our regularization term is one that is as small as possible in order to:

- Prevent overfitting
- Improve generalization
- Reduce variance

In order to find these hyperparameters, we used a technique called GridSearchCV, that fits our logistic regression models with different regularization term values and then performs cross-validation on each of these logistic regression models and then returns the score of each test, as well as the mean of the scores that each logistic regression model obtained throughout the cross validation and the standard deviation between the results from the cross-validation.

Obtaining the optimal Lasso regularization value
We started with a Logistic regression model using a Lasso regularization term and performed a GridSearchCV with the regularization term equal to 1, 0.1, 0.01, 0.001 and 0.0001. The results yielded that the optimal value for the regularization term is between 0.05 and 0.005.

By performing a GridSearchCV between 0.05 and 0.005 we can find the smallest value for the regularization term, that still obtains a high accuracy score. We found the optimal value for the regularization term to be 0.009. After training our model and testing it with our test data set, we saw that a lot of the models weights were reduced to 0. This effectively means that these features with a weight of 0 have no impact on the prediction of our model. We saw that on average after a few cross-validation tests, that only 10 of the features had a weight that was not 0.

Obtaining the optimal Ridge regularization value
Next we performed a GridSearchCV with the regularization term equal to 1, 0.1, 0.01, 0.001 and 0.0001 to obtain the optimal value for our regularization term. The results yielded that the optimal value for the regularization term is between 0.01 and 0.001.

By performing a GridSearchCV between 0.01 and 0.001 we can find the smallest value for the regularization term, that still obtains a high accuracy score. We found the optimal value for the regularization term to be 0.005. After training our model and testing it with our test data set, we saw that all of the models weights were reduced to a value close to 0. This effectively means that each feature has a really small weight, with some of the features having such a small weight, that they will almost always have no effect on our predictions.

Obtaining the optimal Elastic Net regularization value
Lastly when we use Elasticnet as our regularization term, we have to perform GridSearchCV on two parameters, being the regularization value itself and the l1-ratio. The l1-ratio tells the logistic regression what the ratio of Ridge regularization is against Lasso Regression. If we use a l1-ratio of 0.1, then the logistic regression model will use (0.1)L1 + (0.9)L2.
When we apply our GridSearchCV, we obtained that the optimal value for our regularization term falls between 0.01 and 0.001 when we use a l1-ratio of 0.4. Using GridSearchCV again between 0.01 and 0.001 with l1-ratio=0.4, we observe that the most optimal value for our regularization term is 0.009. After training our model and testing it with our test data set, we saw that some of the models weights were reduced to a value close to 0 and the rest of the weights were reduced to 0. This effectively means that some features has a really small weight while other features had no impact on the predictions.

Our model results
We now have 3 logistic regression models that we build and fit to make predictions on the data. These three being logistic regression models using:

- Lasso Regularization with value equal to 0.009
- Ridge Regularization with value equal to 0.005
- Elastic Net Regularization with value equal to 0.009 and l1-ratio equal to 0.4

We performed a few K-fold cross-validation procedures on these three logistic regressions to obtain the following:

K-Fold cross-validation average results				
Type of Logistic regression	cross-validation with K=5	cross-validation with K=10	cross-validation with K=20	cross-validation with K=50
Lasso	0.949	0.949	0.95	0.95
Ridge	0.96	0.961	0.96	0.96
Elastic net	0.961	0.961	0.96	0.96

When performing predictions on the dataset, it is very important that we minimize the following incorrect predictions:

- When we predict that a person has a high chance of having lung cancer, when the person actually has a small chance of having lung cancer.
- And when we predict that a person has a low chance of having lung cancer, when the person actually has a high chance of having lung cancer.

We had to investigate these two values for all of our logistic regression models. We performed 50 different tests and obtained the following averages:

Type of Logistic regression	Predicted high, is actually low	Predicted low, is actually high
Lasso	1	0
Ridge	0	0
Elastic net	0.15	0

5.4 Random Forest Model(s)

We investigated two types of methods of building Random Forest Classifiers models[9] in order to fit the best possible Random Forest Classifier to our data. The difference in the two trees that we investigate came down to the function they use to measure the quality of a split. We call this a trees criterion. The two types of criterion we used was:

- gini criterion Random Forest Classifier
- entropy criterion Random Forest Classifier

Preventing overfitting
A Random Forest classifier has a lot of hyperparameters that can be tuned in order to prevent overfitting. After doing multiple GridSearchCV's, we came across the best three parameters to tune that prevented overfitting the best and they were:

- min_impurity_decrease: This parameter controls the minimum impurity decrease required for a split to occur. Increasing this parameter prevents the tree from splitting nodes that result in minimal improvement in impurity, thus leading to a simpler tree. It limits unnecessary splits.
- max_leaf_nodes: This parameter limits the maximum number of leaf nodes in the tree. Having a quite small value set as this parameter is ideal, as it prevents the tree from becoming too deep and complex, thereby reducing the risk of overfitting.
- n_estimators: This parameter is the number of decision trees in the forest ensemble. By increasing this, we prevent overfitting, by reducing variance, but we can't use too much decision trees in our ensemble, as too many trees can lead to a high Bias.

Obtaining the optimal hyperparameters for the gini criterion Random Forest Classifier
Firstly we want to determine the most important parameter, that is the min_impurity_decrease parameter. We applied a GridSearchCV on our data and after investigating our findings, we saw that the most optimal value for our min_impurity_decrease parameter is 0.05.
Using this, we were now able to find the most optimal value for the max_leaf_nodes parameter by using GridSearchCV. We found this value to be 6.
Lastly we had to find an optimal value for n_estimators. We want it to be as high as possible in order to decrease variance, but not too high, as it would cause the bias of the model to be too large. After applying GridSearchCV, we found the most optimal value for the n_estimators parameter to be 500.

Obtaining the optimal hyperparameters for the entropy criterion Random Forest Classifier
We again want to determine the min_impurity_decrease parameter's most optimal value first. After applying GridSearchCV, we obtained a most optimal value of 0.1.
Next we used GridSearchCV to obtain the most optimal value for the max_leaf_nodes parameter. After investigating the results, we saw that the most optimal value for this parameter was 6.
Then we had to tune our final hyperparameter, that was n_estimators. After applying a GridSearchCV and investigating the results, we found the most optimal value for this parameter to be 600.

Our Model results
We now have 2 Random Forest Class models that we build and fit to make predictions on the data. These two being Random Forest Classifier models using:

- gini criterion Random Forest Classifier with hyperparameter's:
  - min_impurity_decrease = 0.05
  - max_leaf_nodes = 6
  - n_estimators = 500
- entropy criterion Random Forest Classifier with hyperparameter's:
  - min_impurity_decrease = 0.1
  - max_leaf_nodes = 6
  - n_estimators = 600

We performed a few K-fold cross-validation procedures on each random forest classifier model to obtain the following:

K-Fold cross-validation average results				
Type of Criterion	cross-validation with K=5	cross-validation with K=10	cross-validation with K=20	cross-validation with K=50
Gini	0.97	0.966	0.955	0.97
Entropy	0.98	0.979	0.975	0.973

It is very important to remember what our model tries to predict, thus it is very important that we minimize the amount of incorrect predictions that can have a huge impact on someone. These predictions are:

- When we predict that a person has a high chance of having lung cancer, when the person actually has a small chance of having lung cancer.
- And when we predict that a person has a low chance of having lung cancer, when the person actually has a high chance of having lung cancer.

It is very important for us to investigate these two values as incorrectly predicting one of these can lead to someone losing their life when it could have been prevented, or someone wasting a lot of money to try and prevent lung cancer, when the person actually had a low chance of having lung cancer. We performed 50 different tests on each random forest classifier model and obtained the following averages for both these values:

Type of Criterion	Predicted high, is actually low	Predicted low, is actually high
Gini	1	0
Entropy	0.55	0

6 A Comparison of Model Results

Accuracy of the models

There are numerous things to consider when we compare the models to one another in order to find the best model for making predictions on the data set. The first thing we will be looking at and comparing is the average accuracy scores of each model after performing numerous K-fold cross-validation tests on each model.

K-Fold cross-validation average results				
Model	cross-validation with K=5	cross-validation with K=10	cross-validation with K=20	cross-validation with K=50
Current approach logistic regression	0.919	0.919	0.919	0.919
Logistic regression with Lasso regularization	0.949	0.949	0.95	0.95
Logistic regression with Ridge regularization	0.96	0.961	0.96	0.96
Logistic regression with Elastic net regularization	0.961	0.961	0.96	0.96
Random Forest Classifier with gini criterion	0.97	0.966	0.955	0.97
Random Forest Classifier with entropy criterion	0.98	0.979	0.975	0.973

After obtaining these accuracy results, we can clearly see that all of our models performed better than the current approach model. The Random Forest Classifiers clearly performed the best in terms of accuracy and out of the two Random Forest Classifier models, the one with entropy criterion performed the best. Something we also took note of was the Logistic regression with Ridge regularization performed the best out of all of the Logistic regression models.

Important predictions we need to consider

A very important aspect to consider when we make predictions of the chance a person has of having lung cancer is the implications of an incorrect prediction. The second most important prediction to take account of is when we predict someone to have a high chance of having lung cancer, when in actuality, the person has a low chance of having lung cancer. This can be a really bad prediction, because of how much money it can cost a person to go for examinations or treatments and how much time this person would waste.

The most important prediction we need to take account of is when we predict a person to have a low chance of having lung cancer, when in actuality the person has a high chance of having lung cancer. This prediction could lead to a person not taking the necessary steps to prevent lung cancer and when the person realises that they have lung cancer, it might be too late, resulting in the person losing the battle.

We ran 50 tests on each of the models obtaining both of these predictions for each of the tests and calculated their averages. In order to compare each models results, we summarised it in the table below:

Model	Predicted high, is actually low	Predicted low, is actually high
Current Approach logistic regression	0	0
Logistic regression with Lasso regularization	1	0
Logistic regression with Ridge regularization	0	0
Logistic regression with Elastic net regularization	0.15	0
Random Forest Classifier with gini criterion	1	0
Random Forest Classifier with entropy criterion	0.55	0

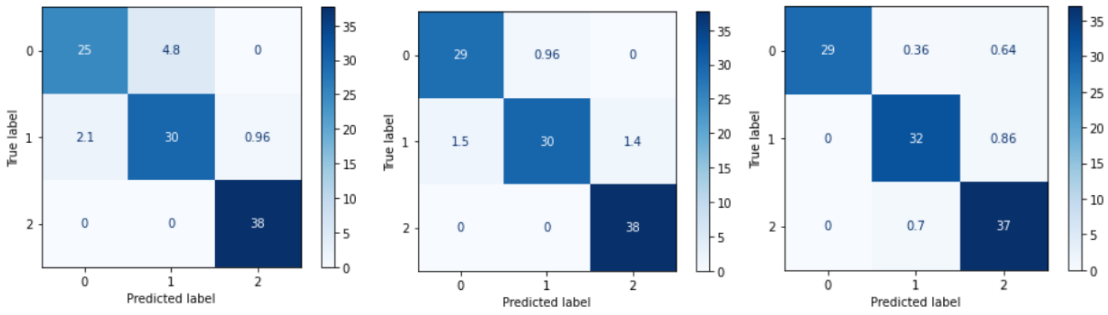
From this table, we can see that the current approach got 0 for both of these predictions, as did our Logistic regression model with Ridge regression that obtained the best score out of all of the Logistic regression models. Our Random Forest Classifiers did not perform as well, although they also didn't get one wrong prediction for the most dangerous prediction, they still did not perform as good as the logistic regression with Ridge regularization model when predicting a high chance of having lung cancer, when the person actually has a low chance.

Comparing our best models

From here on out we could clearly see that our Logistic regression with Ridge regularization model and our Random Forest Classifier with entropy criterion performs the best out of all the other models. So from here on out, we will only be comparing them against the current approach.

Confusion matrix comparisons

The next step we took in comparing our models was that we again ran 50 tests on each model in order to compare all of the models correct and incorrect predictions. We again obtained the average of each of these values to obtain:



(a) Current approach Logistic regression

(b) Logistic regression with Ridge regularization

(c) Random Forest Classifier with entropy criterion

Figure 8: The average confusion matrices after 50 tests.

We also want to consider the weighted average of our precision, recall and f1-score of the 50 tests we ran on each model. To do this, we obtained the averages of each of these values from our 50 tests. This weighted average tells us how well the model performed when considering all of the classes and not just one class, thus giving us a weighted average. These weighted averages were:

Weighted averages			
Model	Precision	Recall	F1-score
Current approach logistic regression	0.923	0.9216	0.9216
Logistic regression with Ridge regularization	0.962	0.9614	0.9614
Random Forest Classifier with entropy criterion	0.9744	0.9744	0.9744

After considering all of these, we have concluded that the Logistic regression with Ridge regularization model is the best model to use for predictions on this dataset. Although the Random Forest does obtain a higher accuracy than the Logistic regression, the trade off between more accuracy and the severity of the predictions is just not worth it. Risking peoples life's, time and money is not worth a 0.01 higher accuracy. The weighted averages also reduces a lot for the random forest classifier, so there is a class that gets predicted a little too wrong and we can see that class is the high chance of having lung cancer class.

Therefore we choose the Logistic regression with Ridge regularization is our best model for making predictions on this dataset.

7 Actionable Insights

Following the results just shown, how does our model add to the problem statement? How does it improve upon the current state of lung cancer predicitons?

For starters, our model should theoretically perform really well on unseen and new data. This is due to our models outstanding performance metrics, which includes high accuracy, precision, recall, and f1-score. Additionally, it maintains a balance between bias and variance, ensuring robustness without overfitting.
In terms of lung cancer prediction, as stated before, minimizing false positives and false negatives is vital when dealing with the lives of prospective patients and as shown previousy, our model is also capable of minimizing these results.
Essentially, in the field, this model serves as a cost effective and non-location dependant means of helping people in need in privileged and underprivileged regions.

To conduct tests, all that is theoretically needed is a medical professional who can conduct a set questionnaire to a prospective patient, input the responses into a dataset (with expert knowledge) and then run the model. Once this is done and based on the outcomes of the prediction, said medical professional can then refer a patient to their next steps in diagnosis. This can be done anywhere at anytime. The model is able to run on almost any basic computing device, making it accessible to many different areas of the world.

As stated in the problem statement, early detection is key when it comes to maximizing survival rates of lung cancer. There are many people around the world who might have a hunch, suspecting that there is a problem with their lungs, but do not have the time, energy or money to invetsigate the issue further. We believe that this model can be used as a tool to bridge this gap.

Lastly, applying our model and its application to South Africa:
It is well known that South Africa has dense populations living in highly polluted areas as well as a large amount of smokers. South Africa is also riddled with underemployment and poor living conditions in many areas. These are the areas where our model will be most effective. We believe that if implemented correctly, our model could have real world application and improve the current state of the medical field in regions such as South Africa.

8 Reflection

This project has governed an in depth understanding, analysis and improvment of a problem that was not ours to begin with. It forced us to critically engage with and critique every possible angle, consideration and methodology one should take or think about when building a predictive model from scratch. It also taught us the importance of understanding the problem statement at hand and how closely one should work with said problem statement and manipulate a data set in such a way that one maximizes the predictive power of a model whilst also maintaing a good balance of bias and variance.
Another important lesson for us was learning the true meaning of the word 'overfitting'. When we first ran the *current approach* an observed the accuracy value of 90.3%, we did not think that it would be possible to improve. However as soon as we noted that the author used minimal to no regularization techniques as well as a poor variable selection method, we saw great room for improvment in that regard and based our project soley on trying to replicate a similar result of that of the authors with the added and improved data cleaning, data pre-processing, regularization techniques and many more. Once we started to understand the problem at hand in great detail and put our focus into minimizing false positive and false negative rates (as this is incredibly important for the classification of lung cancer patients), we found a substantial increase in all models built in terms of the following metrics:

- A substantially less overfit model
- Higher predictive accuracy
- Increased recall, precision and f1-score
- A better model that has a good bias variance rade-off
- And overall, a great model for classifying potential lung cancer patients

However, it goes without saying that there were many bumps in the road. If we were to start over, we would've skipped preliminary steps such as LDA and gone straight to lasso and ridge regression. One vital mistake we made early on, was not implementing a proper cross-validation system to test our model on many different subsets of the training data. We learnt the hard way of the importance of doing such.
If we had more time, we would've liked to explore support vector machines (SVM's). A SVM has many moving parts and is incredibly complex. And as aspiring data scientists, learning how and why these models work is vital when it comes to applying them to real world problems.

We are pleased to report significant personal and professional growth as data scientists through this project and what we have done. We are satisfied with our achievements and eagerly anticipate future opportunities in the ever evolving field of data science.

Appendix: References

[1] Cancer Association of South Africa (CANSA), "Lung cancer," https://cansa.org.za/lung-cancer/, 2024, accessed: 2024-03-16.

[2] Roche, "Facts about lung cancer," https://www.roche.com/stories/facts-about-lung-cancer, 2024, accessed: 2024-03-16.

[3] K. P. Anagha, "Lung cancer prediction: Logistic regression model," https://www.kaggle.com/code/anaghakp/lung-cancer-prediction-logistic-regression-model, 2024, accessed: 2024-03-16.

[4] "Kaggle: Your home for data science," https://www.kaggle.com/, accessed: 2024-03-16.

[5] TheDevastator, "Cancer patients and air pollution: A new link," https://www.kaggle.com/datasets/thedevastator/cancer-patients-and-air-pollution-a-new-link, 2024, accessed: 2024-03-17.

[6] J. Smith and J. Doe, "Impact of air pollution on lung cancer rates in nonsmokers," https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3574203/, 2023, accessed: 2024-03-17.

[7] V7 Labs, "A comprehensive guide to confusion matrices," https://www.v7labs.com/blog/confusion-matrix-guide, 2024, accessed: 2024-03-20.

[8] RITHP, "Logistic regression and regularization: Avoid overfitiing and improving generalization," https://medium.com/@rithpansanga/logistic-regression-and-regularization-avoiding-overfitting-and-improving-generalization-e9afdcddd0
2023, accessed: 2024-03-18.

[9] A. Shafl, "Random forest classification with scikit-learn," https://www.datacamp.com/tutorial/random-forests-classifier-python, 2023, accessed: 2024-03-25.