Logistic Regression and Deep Neural Networks in Predicting Lung Cancer, a Comparison

Hayden White
Department of Computer Science
University of South Carolina Upstate
Spartanburg, South Carolina, USA
September 10, 2022
(864) 279 - 0920

mhwhite@email.uscupstate.edu

Abstract
Cancer is a disease in which normal functioning cells in the body become abnormal and divide uncontrollably. A cluster of these cells is called a tumor, and when a tumor grows and spreads into other tissues of the body the disease can become life-threatening. One of the most common cancers for women is breast cancer and for men, prostate cancer. For both men and women, colon cancer and lung cancer are also common. Lung cancer in particular is the most common type of cancer diagnosis in the world and one of the leading causes of preventable death. This makes it extremely important to confirm a positive diagnosis of lung cancer as early as possible. There are physical symptoms associated with lung cancer such as coughing, wheezing, fatigue, and many more. One method of determining a diagnoses of lung cancer is through machine learning algorithms that input a key list of the physical symptoms associated with the disease that an individual patient might be experiencing, such as the symptoms listed previously, and outputting a likelihood that the culmination of these symptoms in the patient is indicative of a positive case of lung cancer. This practice naturally invites the question of which machine learning algorithm is most accurate and efficient in determining such an intimate and devastating diagnosis as lung cancer. The following research seeks to answer this question by comparing two reputable machine learning models with an established footprint in modern academia: Logistic Regression and Deep Neural Networks.

Keywords
Passive Smoking, Hemoptysis, Dyspnoea, Mammography, Mesothelioma, Comorbities, Machine Learning, Deep Learning, Neural Network, Deep Neural Network, Logistic Regression, Linear Regression, Backpropagation, Stochastic Gradient Descent, Partial Derivatives, Multivariable Calculus

1. Introduction
The epidemiology of Lung Cancer finds its roots in the first half of the twentieth century, sometime around the year 1930. Before this period, Lung Cancer was an extremely rare disease, despite tobacco consumption having been popularized for hundreds of years prior to this point. The prevalence of the disease culminated in the mid-twentieth century when Lung Cancer became the leading cause of cancer death for males. For females it did not take much longer until a sharp rise in lung cancer rates starting in the 1960s to the 2000s propelled lung cancer to become the most frequent cause of female cancer mortality as well [1]. The epidemic of lung cancer among females occurred later and never peaked as high as it did among males because the prevalence of smoking tapered off at substantially higher levels among males around the late eighties to early nineties [1].

Today, we can readily see that past trends in the rise and fall of cigarette smoking has a direct correlation to the fluctuations in the prevalence of positive cases of lung cancer. As previously stated, tobacco had been widely used for hundreds of years, however the modern epidemic of lung cancer directly followed the introduction of mass manufactured cigarettes with addictive properties [1]. An increasing lifespan resulting from the industrial revolution, the sustained exposures of inhaled carcinogens to the lungs, and the addition of new exposures to etiologic agents in cigarettes lead to lung cancer taking the twentieth century by storm. German scientists in the 1930s and early 1940s facilitated some of the earliest research on the connection between lung cancer and smoking and by the early 1950s studies in Britain and the United States corroborated this strong association between cigarettes and the risk of lung cancer. In the year 1964, the United States Surgeon General claimed all evidence was sufficient enough to support the conclusion that cigarette smoking indeed caused lung cancer. Compared to those who have never smoked before, active smokers have an increased risk of developing lung cancer by a factor of twenty. However, while the pattern of lung cancer occurrence is reflective of smoking trends, the rates of occurrence of lung cancer lag behind smoking rates by roughly twenty years [1].

While cigarette smoking is irrefutably the leading cause of lung cancer, other factors that can increase the risk of developing lung cancers also exist. The involuntary inhalation of tobacco smoke, referred to as passive smoking, has been proven to cause lung cancer. Outdoor air pollution, including but not limited to combustion-generated carcinogens, is considered to increase the risk of lung cancer among urban populations. There is also the concern that in some cases indoor air can carry many respiratory carcinogens such as asbestos and radon. Lastly, there is evidence continuing to emerge that genetic determinate can increase the risk of lung cancer [1]. Despite the numerous other factors that can contribute to the development of lung cancer, active smoking is still responsible for 90% of all lung cancer cases while outdoor air pollution only accounts for about 2% or less of lung cancer cases. One last notable contributor to lung cancer causation is dietary factors, which have been hypothesized to account for approximately 10 to 30% of the lung cancer burden [1].

Lung cancer prevalence tends to be particularly higher among developed nations in Europe and North America and is less common in the developing nations of South America and Africa. Within those developed countries, the occurrence of lung cancer by race and ethnicity makes the disease relevant for those concerned with the health of minorities. Lung cancer occurs 50% less frequently among white men compared to African-American men [1]. Socioeconomic status is also a major determinate in the risk of lung cancer. In the United States, lung cancer mortality rates among white men is lowest in the Northeast and highest in the Southeast. Low socioeconomic status is associated with an unfavorable profile of several determinants of increased lung cancer risk such as exposure to inhaled agents in the workplace, diet, the prevalence of smoking, and the general environment. The Center of Disease Control has documented the deaths caused by smoking-related lung cancer in the United States. For the year 1990, the United States tallied the most deaths in the world with a body count of 127,000. This figure is multiplied when accounting for the rest of the developed world, whose smoking-related fatalities amassed to 457,371 in the year 1990.

Despite a general decreasing trend of lung cancer in the developed world, some countries are experiencing an increase in lung cancer rates. A huge burden of lung cancer cases is predicted for China, which has now become home to one third of the world's smokers. Their smoking-related death count is predicted to amount to several million by the middle of the twenty-first century [1]. Since lung cancer is still the most common diagnosed cancer worldwide, and on the rise in so many countries, the demand for highly accurate models that can predict positive diagnoses of lung cancer are of utmost importance. The earlier lung cancer can be detected, the sooner treatment can begin to combat it, and one of the most cutting-edge facets of medicine today is the development of machine learning algorithms that detect the prevalence of cancer in the human body. In this research, I will be comparing two popular machine learning algorithms: Logistic Regression and Deep Neural Networks. We will investigate the differences in implementing each model, compare the efficiency and complexity of both algorithms, and also the the accuracy of both algorithms in predicting Lung Cancer.

2. Literature Review

Surprisingly, despite machine learning algorithms having been used to create predictive models for decades now, applying them towards cancer diagnosis predictions is a relatively new field with few published studies. Luckily, the studies that do exist go into great detail their methodology and results for applying machine learning algorithms to create cancer-related predictive models. A variety of of algorithms including Decision Trees, Support Vector Machines, Bayesian Networks, and Artificial Neural Networks have been applied in cancer research for the creation of predictive models that have resulted in accurate and effective decision making [3]. There is also a large amount of cancer data that has been made available to the medical research community in which data scientists can find relationships and patterns between complex data to predict future outcomes of a cancer types [3]. The types of machine learning algorithms used effects the performance of a specific model, and each algorithm has their own identifiable advantages and weaknesses. In the last few years, the accuracy of cancer predictions through the use of machine learning models has improved by 15-20%, however these studies will require more adequate validation through larger data samples [3].

A popular machine learning model used in cancer prediction as previously stated, is Artificial Neural Networks. ANNs are supervised learning algorithm, or in other words, they take a labeled set of training data so that they can map or estimate input data to a desired output. The task of a supervised learning algorithm is to categorize data into a set of finite classes. Usually with cancer prediction models we are referring to a binary classification output. For example, an ANN that is modeled to predict the probability that a tumor is or is not malignant (0 for "no," 1 for "yes"). This would also be considered a regression problem, in which a learning function maps input data to a real-value variable [3]. Artificial Neural Networks serve as the gold standard in several classification problems used in cancer prediction, but they also suffer from some drawback. Their layered structure can be time-consuming and lead to poor performance. Also, if an ANN does not work properly in its cancer prediction process, all the hidden layers make it nearly impossible to detect why the neural network did not perform as well as anticipated [3].

When developing an ANN model, or any machine learning algorithm for that matter, it is important to keep data-related issues in mind that can occur when processing a large amount of complex data. This can include issues such as duplicate or missing data, outliers, noise, or data that is considered biased-unrepresentative. There are a few techniques for dealing with these issues in a stage referred to as "data preprocessing" that include dimensionality reduction, feature selection, and feature extraction. Machine learning algorithms learn better when the dimensionality is lower, making dimensionality reduction an important step in reducing noise and providing a more robust model. Feature selection coincides with dimensionality reduction as it is the process of selecting a new feature set that is a subset of the old feature set. Finally, feature extraction creates a new set of features from the initial set that captures all the significant information in a dataset [3].

A successful disease prognosis is dependent on the quality of a medical diagnosis, but a prognostic prediction should ideally take into account more than a simple diagnostic decision. Three predictive tasks should be of concern when reckoning with cancer prediction: the prediction of cancer survival, the prediction of cancer recurrence, and the prediction of cancer susceptibility. For the first case, the main objective is predicting a survival outcome that is disease specific or overall survival after cancer treatment

begins. For the last two cases, the objective is finding the likelihood of developing a type of cancer and the likelihood of redeveloping that type of cancer after partial or complete remission [3]. Machine learning algorithms such as ANNs and Decision Trees have been used in these types of prediction models for nearly three decades, and a growing trend in the last ten years has been the use of supervised learning algorithms specifically towards cancer prediction and prognosis. Physicians conclude the integration of features such as weight, diet, age, family history, high-risk habits, and exposure to environmental carcinogens play an important role in predicting the development of cancer. The expression of certain genes, cellular parameters, and molecular biomarkers are also proven to be informative cancer prediction indicators [3].

One study looks into developing decision making tools that can discriminate between malign and benign breast cancer tumors. When developing these prediction models, risk stratification is important and other existing studies use computer models using techniques such as ANNs to asses the risk of breast cancer patients. These neural networks are distinguishing between benign and malignant tumors through mammography findings, or the use of x-rays to diagnose and locate tumors in the breast. These ANN models use a large amount of hidden layers which work better than neural networks with a smaller number of hidden nodes. For this study in particular, the data collected and used in these models consisted of 48.774 mammography findings as well as tumor characteristics and demographic risk factors. Note, radiologists reviewed all of the mammography records [3]. This data was then inputted in the ANN model. Performance was estimated by ten-fold cross validation. The calculated Area Under the Curve of this model gave a 0.965 accuracy following its training and testing, which the authors claim signifies this model can accurately estimate the risk assessment of breast cancer patients. They also concluded the most important factors used to train the neural network were the mammography findings with tumor registry outcomes [3].

The second study we will analyze identifies a combination of early predictive symptoms and sensations indicative of primary Lung Cancer. An e-questionnaire comprised of descriptors of these symptoms and sensations were administered for patients suspected of having Lung Cancer, each descriptor requesting a binary response of “yes” or “no.” This data was then inputted into a machine learning multivariate regression algorithm called OPLS, or Orthogonal Projection to Latent Structures [4]. The goal of this study was to create a model that helps identify early risk symptoms and sensations of Lung Cancer that can flag individuals for screening and early detection. The early identification of these lung cancer symptoms will greatly impact overall lung cancer mortality resulting from greater survival in early-identified stages [4].

Data was collected from a total of 1200 patients, however 530 of those patients were excluded due to not meeting inclusion criteria, declining to participate, or for other reasons not specified. This resulted in a full sample of 670 patients, which was then filtered again by excluding patients had another or earlier cancer diagnosis, mesothelioma, or other factors. This resulted in a total dataset of 506 patients. All data was anonymized to protect the privacy of the study participants [4]. The e-questionnaire consisted of many symptoms starting with Background (smoking habits, comorbidities, sociodemographic characteristics, etc.), Breathing Difficulties, Phlegm/Expectorates, Pain/Aches/ Discomfort, Fatigue, Voice Changes, Appetite/Eating/Taste Changes, Olfactory Changes, Fever/Chills/Sweating, and Other Changes adding up to 342 total potential items and 285 descriptors, however, after several tests with the OPLS model, these descriptors were narrowed down to 70 variables in total (63 descriptors and 7 background variables) [4].

The 70 variable model resulted in the most accurate model performance with an Area Under the Curve of 0.767, a sensitivity of 84.8%, and a specificity of 55.6%. The author claims this was the first study to their knowledge to utilize an e-questionnaire given to individual patients suspected for Lung Cancer that asks for self-evaluation of pre-diagnostic descriptors of symptoms for associated with Lung Cancer. This questionnaire was combined with a cutting-edge multivariable machine learning analysis of multi-dimensional data to find how different combinations of variables perform in predicting Lung Cancer [4]. In addition to active smoking being the most recognized risk factor, chest pain, cough, dyspnoea, and hemoptysis were also important contributors. The study does recognize, however, that patient recall bias is a potential inhibitor to the accuracy of the prediction model. They also recognize that a larger sample would aid in discovering the importance the selected descriptors, and that their model needs to be tested against the general population in order to properly evaluate its validity as a tool in determining patients to flag as at-risk for Lung Cancer [4].

Now let's discuss Logistic Regression, one of the models I will be comparing in this research. Logistic Regression is a learning algorithm that estimates the association of one or multiple predictor variables with a binary output variable. We call the output variable “binary” because it can only take the form of two states (on or off, yes or no, 1 or 0, etc.). More directly, Logistic Regression is used to estimate the probability of an outcome happening depending on the value of the input variables [5]. This probability has a sigmoidal relationship with the input variables, conveniently constraining the output between 0 and 1. Logistic Regression can easily accommodate for more than one independent variable, allowing us to analyze the relationship each variable has on the binary outcome. This is one of the reasons Logistic Regression is one of the most commonly used predictive models for dichotomous outcomes in medicine [6]. For our research, this means we can analyze the effect each individual symptom associated with Lung Cancer has on the likelihood of the patient being diagnosed with Lung Cancer.

One of the major advantages of Logistic Regression is that the exponentiated logistic regression slope coefficient can be easily interpreted as an odds ratio, indicating how much the odds of an outcome can change versus a reference category (when dealing with categorical input variables, specifically) [5]. The second model we will be comparing, a Neural Network, also has advantages. They require less formal training statistically, they have the ability to implicitly detect complex nonlinear relationships between input and output variables, the ability to detect all possible interactions between predictor variables, and also the availability of several training algorithms [6]. Neural networks also have disadvantages, one of them being the “black-box” nature of them that makes it hard to pinpoint exact locations in the model that are problematic. Other disadvantages are the computational burden, proneness to overfitting, and the empirical nature of model development [6].

3. Methodology

This section will explore an in-depth overview of both machine learning algorithms used in this study: Logistic Regression and Deep Neural Networks.

3.1 Logistic Regression

Regression models are an important research method when investigating the different associations between an output variable and multiple independent variables, commonly referred to as explanatory variables, predictors, or covariates. One of the most frequented regression methods is Linear Regression which is implemented for an alleged continuous outcome that assumes its relationship with various independent variables produces a straight line [7]. These models can be performed with a single independent variable, but it is usually more effective to evaluate the changes in multiple independent variables simultaneously in how they affect the outcome variable. There is a common formula in Linear Regression modeling as follows (note that b represents the greek 'beta' notation):

$$Y = b_0 + b_1X_1 + b_2X_2 + \ldots b_iX_i \qquad (1)[7]$$

For the components of this function, Y is the continuous output variable, b_0 is the intercept with the dependent axis, and $b_1X_1 + b_2X_2 + \ldots b_iX_i$ is the set of independent variables weighted by their beta coefficient. The beta coefficients provide the slop of the regression curve as the outcome increases with each one unit change in the independent variables [2].

While the Linear Regression model is not applicable for the research presented in this study, it is important to preface it before examining its close relative: Logistic Regression. As stated previously, Logistic Regression is used to estimate a probability, binary outcome as the estimation of a single or multivariate input [5]. The key difference here is that the output in Logistic Regression is not continuous, but rather a binary output of either 0 or 1 or some probability in between. This will be especially useful for creating a predictive model for Lung Cancer, since we are not looking for a continuous answer but simply a probable likelihood of a positive or negative diagnosis.

Note that the use of multiple independent variables is also more productive for Logistic Regression as we can asses their unit changes simultaneously and the affect these unit changes have on the binary outcome. Assessing Logistic Regression and specifically these unit changes in relation to the output can be modeled with he following formula:

$$Y = \frac{e^{\beta_0+\beta_1X_1+\beta_2X_2+\ldots\beta_iX_i}}{1 + e^{\beta_0+\beta_1X_1+\beta_2X_2+\ldots\beta_iX_i}} \qquad (2)\ [7]$$

Within this equation, notice that the configurations of the intercept variable as well as the independent variables and their beta coefficients are identical to their Linear Regression counterpart, except here these expressions are the exponent of base e of the natural logarithms. The output Y is now constrained to 0 as its lowest possible output and 1 as its highest possible output, hence the naming convention 'Logistic Regression.' In other words, the independent variables are expressed in the logit scale as opposed to a linear model, and the output Y is the estimated probability of a binary outcome as opposed to a continuous outcome [7]. The logit scale naturally transforms the original Linear Regression formula to produce the natural log of the odds in the first category of outcomes (Y) versus the second category (1-Y) and can be expressed in similar terms to the Linear Regression formula through the following, simplified natural log expression:

$$\text{Logit}(Y) = \ln(Y/(1-Y)) = b_0 + b_1X_1 + b_2X_2 + \ldots b_iX_i \qquad (3)[7]$$

Logistic Regression creates what is commonly referred to as a sigmoidal curve, which plots our output values within the restraints of the lowest possible outcome 0 and the highest possible outcome 1:

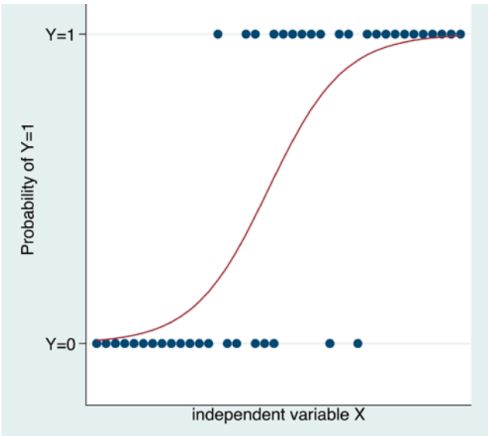


Figure 4. Sigmoid Curve [5]

Lastly, it is important to note the visual implication of this curve which was briefly touched on previously. While the constraints for the output Y are present, it is still possible for a culmination of input variables to output a value along the sigmoid curve that is not at Y=0 nor Y=1. There can be intermediary outputs as well. This leads to an important factor when designing a Logistic Regression model: the threshold. This is an arbitrary parameter designated by the model creator and indicates whether an output is assumed to be a 0 or 1. For example, if the threshold of a Logistic Regression model is 0.5, then any output that falls below 0.5 is assumed to be a 0 and any output above this threshold is assumed to be a 1. The number designated as the threshold can remedy or create any possible biases in the model and also effect the model's accuracy, so it is extremely important to choose this number carefully.

3.2 Deep Neural Networks

A neural network, artificial neural network, or deep neural network is a type of supervised machine learning algorithm, meaning the network is trained with pre-existing, labeled data before the model attempts to produce outcomes on its own. A neural network is best visualized as a simulation of the human brain, as it consists of "neurons" that take in, process, and output information to other neurons they have built connections with. A neural network consists of multiple layers, and each layer can be thought of as a grouping or level of neurons. There will always be one input layer, one output layer, and one or more intermediary hidden layers. The amount of neurons in the input layer is directly associated with the number of inputs from a single set or row of

labeled data from the dataset. The amount of neurons in the hidden layer is decided by the model creator(s) and while there are several methods for determining this number, it is largely the result of testing which combination of neurons and the number of hidden layers produces the most accurate, well-fitted model. Lastly, the amount of neurons in the output layer is dependent on the type of problem that needs to be solved. A classic example of a neural network problem involves recognizing hand-written digits, so here the output layer would consist of ten neurons, each one representing a digit from 0-9 [8]. A visual representation of a neural network is depicted as follows:

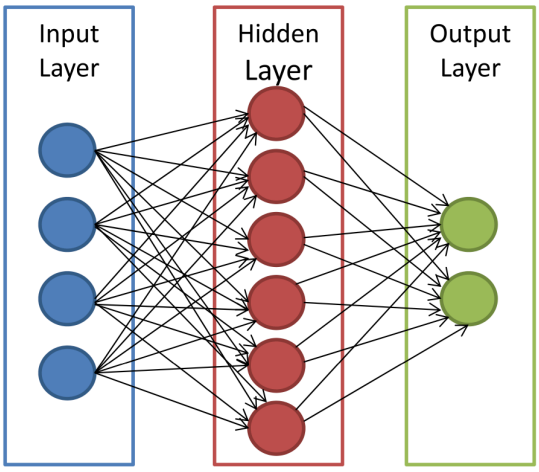


Figure 5. Neural Network Visualization [3]

Each neuron, sometimes called a sigmoid neuron, within the neural network consists of its own function composed of an input, a weight (similar to the beta coefficients in Logistic Regression) and a bias. The output of this function is then fed as the input into the next neuron(s). It is important to note that the input(s) to a single neuron do not have to be either a 0 or 1, but can consist of a number in-between such as 0.458. Sigmoid neurons are effective because they can be modified so that small changes to their weights and bias cause small changes to their output [8]. This function is reminiscent of a simple linear function because it takes on the form w*x + b = output, where w is the weight coefficient, x is the input, and b is the bias. A simple depiction of a sigmoid neuron can be shown as follows:

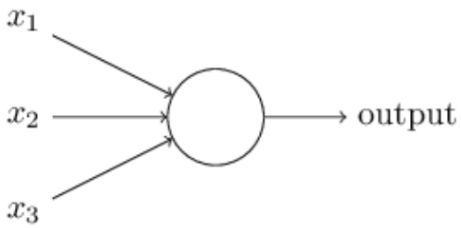


Figure 6. Sigmoid Neuron Visualization [8]

A neural network in which the output from one layer of neurons is the input for the next layer of neurons is referred to as a feedforward network, meaning there are no loops and information is never fed back, only forward [8]. As stated previously, a neural network needs a dataset to learn from, which is commonly called a training set. This data will include the correct output values so that as the network is processing information it knows how to associate what combinations of input data result in which known output, hence the name "supervised" machine learning. This feature is important because it allows for the creation of a cost function, or quadratic cost function, so the output of the neural network can be directly compared to a correct output and the error in the network output, commonly referred to as "loss," can be calculated. This cost function is a function of a predicted output and a correct output, which in themselves are a function of multiple input variables. So one of the problems a neural network attempts to solve is what small changes to the weights and biases of each individual neuron results in an output that produces the lowest possible cost, creating a more accurate model. This naturally leads into the realm of multivariate calculus, and more specifically an algorithm known as Stochastic Gradient Descent. A neural network can consist of innumerable inputs, but when discussing gradient descent it is best to visualize the model in three-dimensions so that a single output of the network can lie on a three-dimensional plane. In essence, the SGD algorithm is designed to take the resulting output, a single point on the plane, and decide what small changes can move the output down the gradient of the plane towards some local minimum value, which reduces the cost function [8]. A visualization if this algorithm can be seen as follows:

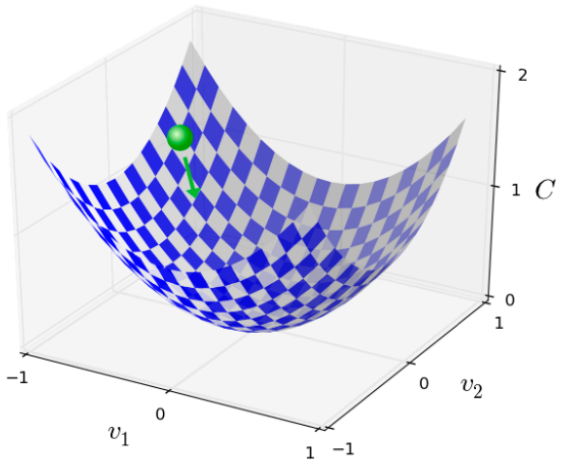


Figure 7. Visualization of Stochastic Gradient Descent [8]

Here, the green dot represents an output of the network, and the lowest point on the plane, the local minima, represents the location that an ideal output would produce the lowest possible cost. The gradient of the function is a vector that points in the direction of steepest ascent, and Stochastic Gradient Descent moves the output by a single step size in the direction opposite of the gradient, or the direction of steepest descent and towards the local minima.

Lastly, and arguably the most important concept when discussing neural networks is an algorithm called Backpropagation, which is a much faster algorithm that produces the gradients as discussed previously. The goal of Backpropagation is to compute the partial derivatives of the Cost function with respect to the weights and biases in the network and generate the possible changes to these

weights and biases that will reduce the overall cost. There are four essential functions associated with the Backpropagation algorithm. These functions are depicted in the following diagram and an explanation for each function shall be provided shortly after:

Summary: the equations of backpropagation

$$\delta^L = \nabla_a C \odot \sigma'(z^L) \quad \text{(BP1)}$$

$$\delta^l = ((w^{l+1})^T \delta^{l+1}) \odot \sigma'(z^l) \quad \text{(BP2)}$$

$$\frac{\partial C}{\partial b^l_j} = \delta^l_j \quad \text{(BP3)}$$

$$\frac{\partial C}{\partial w^l_{jk}} = a^{l-1}_k \delta^l_j \quad \text{(BP4)}$$

Figure 8. Equations of Backpropagation [8]

The first formula, BP1, is an equation for the error in the output layer of the neural network. The second formula, BP2, is an equation for the error in terms of the error of the following layer. Already these first two layers are indicative of the name "back propagation." The algorithm starts by first finding the error of the output layer and then works *backwards,* using the result of the first equation as a component of the second equation. More simply, first the output layer is evaluated, then the layer next to the output layer, and then the next layer and so on. The third formula, BP3, is an equation for the rate of change of Cost with respect to any bias in the network. The fourth and final formula, BP4, is an equation for the rate of change of Cost with respect to any weight in the network [8]. This Backpropagation algorithm is a very effective feature of neural networks and is the reason they are said to have the ability of deep learning.

4. Implementation

This section will briefly discuss the dataset and various technologies that are used in this study.

4.1 Data

This study uses a dataset from the website Kaggle, a popular online database for datasets related to various topics and fields. The dataset chosen for this experiment is called Lung Cancer Detection, which has been updated as of July 2022. It contains 309 rows, each representing an individual patient with anonymity. There are a total of 16 columns or features in this dataset. The first 15 features will serve as the independent variables in this study. Starting with first two features depicting gender and age, respectively, and following that are 13 features which categorize several individually diagnosed symptoms pertaining to Lung Cancer, each consisting of a simple "yes" or "no" answer. The 16th and final column indicates a positive or negative case of Lung Cancer for that patient, which serves as the dependent variable for the study. This dataset is stored in a CSV or "Comma Separated Value" file format.

4.2 Technology

This study takes advantage of a Jupyter Notebooks version 6.4.12 application using a custom Machine Learning Environment containing several packages which include pandas version 1.4.4 for data structures and data analysis, scikit-learn version 1.1.1 for Logistic Regression machine learning, keras version 2.9 for deep learning and regularizers, tensorflow version 2.9.1 for Neural Network machine learning, and matplotlib version 3.5.2 for visualizing data and performance metrics. All of these packages are imported into a Python version 3.9.13 programming environment.

5. Experimental Setup

This section will explore the experimental setup for both the Logistic Regression model and the Deep Neural Network model. Both of these models are further broken down into three unique models respectively due to a discrepancy in the data. After further analyzation of the dataset it was discovered to be biased towards positive cases of Lung Cancer. To be precise, the data contained exactly 270 positive cases of lung cancer and only 39 negative cases of Lung Cancer, which is approximately a 7:1 ratio bias in favor of positive cases. To accommodate this discrepancy, both the Logistic Regression and Neural Network models were categorized into a Full model containing all the data from the dataset, a Half-biased model which contained exactly twice as many positive cases Lung Cancer as negative cases (78 positives and 39 negatives), and an Unbiased model which contained an equal number of positive and negative cases (39 each). All three of theses sub-models will be analyzed and compared to each other as well as the sub-models from the other machine learning algorithm.

5.1 Logistic Regression Setup

The Logistic Regression model began by importing the pandas module so that the dataset csv file could be loaded as an object into the program, and then a preview of the data is printed depicting the first five rows and all 16 features. Initially, Gender is categorized with an 'M' character or an 'F' character for male and female, respectively, and Age is a simple integer of the patients actual age. The remaining symptom categories as well as the dependent variable are tallied as either a '1' indicating the symptom is not present, and a '2' indicating the symptom is present. Age was the only feature that was not altered to fit the model. For the rest of the features, the pandas "replace" method was used as necessary. For gender, every 'M' for male was replaced with a '1' integer and every 'F' for female was replaced with a '0' integer. Similarly, for all of the symptom categories as well as the dependent variable, every '2' was replaced with a '1' integer and every '1' was replaced with a '0' integer. This final step completed all the necessary data alterations to fit the Logistic Regression models.

Next, test_train_data was imported from the scikit-learn module and all of the independent variable features were stored into an array called X. Our Lung Cancer dependent variable was then stored as a single object called Y. The data was now ready to split into training data and test data using the test_train_split method. The test data was set to account for twenty percent of the dataset, leaving the remaining eighty percent for training, and a shuffle-state parameter of 42 was selected to ensure proper shuffling of the data to avoid biases before splitting.

Finally, the Logistic Regression module was imported from scikit-learn and stored in an object called logreg. The maximum iterations parameter was set to 1000 and the solver algorithm selected was 'lbfgs' which stands for Limited-memory Broyden–Fletcher–Goldfarb–Shanno, an algorithm known to perform better for smaller datasets. First, the logreg object was trained using the fit method and passed the X_train and Y_train sub-datasets, and then was tested using the predict method and passing the X_test sub-dataset as it's parameter and stored into a Y_pred object. Lastly, the accuracy_score module was loaded from scikit-learn to determine the accuracy of the model by evaluating the Y_test and Y_pred datasets in a confusion matrix. Lastly, the confusion_matrix and plot_confusion_matrix modules were imported from scikit-learn to visualize the confusion matrix of all three sub-models.

5.2 Neural Network Setup

The initial setup of the Neural Network model is very similar to the Logistic Regression model. Pandas was imported so that the lung cancer dataset could be loaded into an object, and then again a preview of the first five rows of data was printed. The data was still categorized with the same entries, so all of these also had to be altered to fit the model. The Gender feature was changed to either a 1 or 0 to represent male or female, respectively, and all of the symptom features as well as the dependent variables were altered to a 1 or 0 to represent a "yes" (2) or a "no" (1), respectively. However, the age category was also changed for the Neural Network model. Instead of leaving these integers the same, they we're constrained to a maximum value of 1 and a minimum value of 0. This adjustment was necessary because, as discussed previously, sigmoid neurons need to accept inputs that are less than or equal to 1 or greater than or equal to 0.

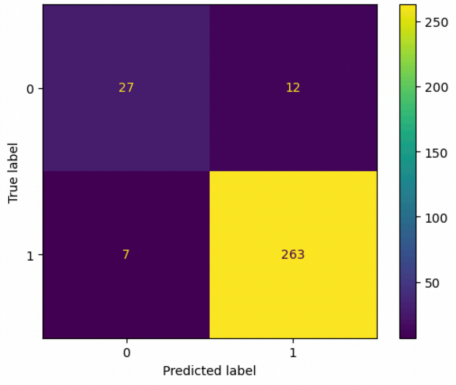
With the data properly altered to fit the model, it was ready to split into a X dataset object containing all the independent variable features and a Y dataset object containing all the dependent data. Then the preprocessing model was imported from scikit-learn so that the age feature adjustment as discussed could be accounted for, and then a new better fitted X_scale independent dataset was printed so these changes are visible. Similar to the Logistic Regression model, test_train_split was then imported from scikit-learn so that the X and Y datasets could be split. This time, seventy percent of the data was used to train the network and thirty percent of the data was stored into an X_val_and_test and Y_val_and_test object for both testing and validation. With both these objects containing validation and test, they were further split in half respectively to create individual test and validate objects for both X and Y. For the full model, this resulted in a X_train object of 216 rows and 15 columns, a X_val object of 46 rows and 15 columns, a X_test object of 47 rows and 15 columns, a Y_train object of 216 rows, a Y_val object of 46 rows, and a Y_test object of 47 rows.

The properly splitter data now enabled the creation of the architecture of the network as well as it's training. The Sequential module as well as the Dense module were imported from the tensorflow.keras library. The Sequential module is used to setup the overall architecture of the Neural Network while Dense module is used to create each individual layer of the module. The input layer was set to 15 neurons since the dataset contains 15 independent features. Then two hidden layers are created, each containing 16 neurons with an activation parameter set to 'relu' to achieve nonlinear transformation of the data and increased training efficiency. The output layer was denoted with a single neuron and its activation parameter was set to 'sigmoid.'

Next, the Neural Network model had to be configured to certain attributes: an algorithm for optimization, an appropriate loss function, and the evaluations metrics this experiment is keeping track of. All of these attributes were configured using the compile method. The algorithm optimizer parameter was set to 'sgd' which stands for the Stochastic Gradient Descent algorithm previously discussed in the Methodology section. The loss function parameter was set to 'binary_crossentropy' because the model requires a loss function for sigmoidal outputs, and the evaluation metric parameter was set to 'accuracy' as this is the metric that is compared with the Logistic Regression model. With the architecture of the Neural Network finished, it could now be trained with the X_train and Y_train objects with a batch size of 4 and number of epochs set to 100, as this was the configuration when paired with this specific architecture resulted in the highest accuracy in testing. The valuation objects for both X and Y were also passed as parameters. After training, the model was tested with the fit function and passed the X_test and Y_test objects. Finally, matplotlib was imported so that all three sub-models could be visualized in terms of both accuracy and loss.

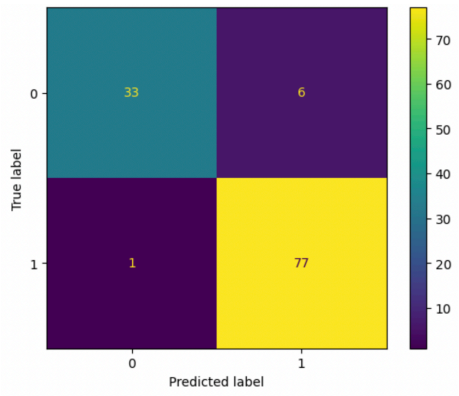
6. Results Analysis

5.3.1 Logistic Regression Full Model



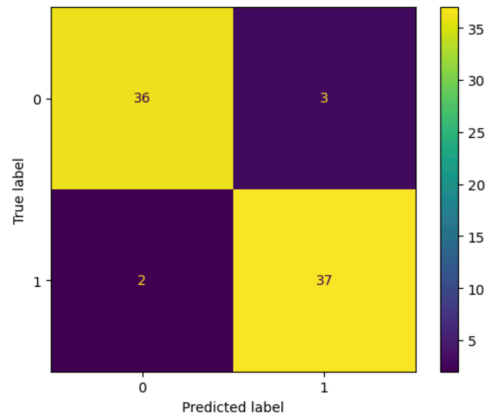
The Logistic Regression full model proved to be the most accurate, with 263 true positive predictions and 27 true negatives, with a total accuracy of 96.77 percent. Note the false positives and false negatives are only at 12 and 7, respectively. This trend of low false results will continue in the next two Logistic Regression models.

5.3.2 Logistic Regression Half-Biased



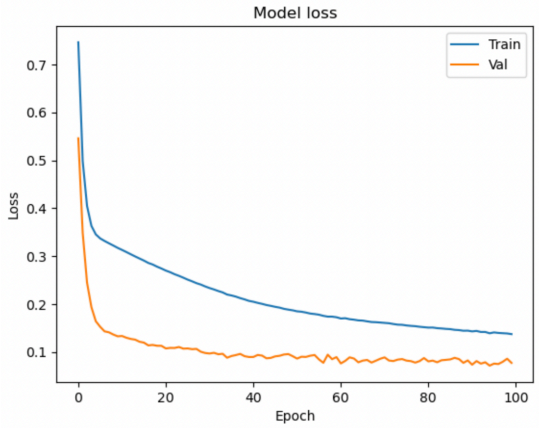
The Logistic Regression Half-Biased model is also very accurate, though slightly less accurate than the full model at 95.83 percent. Here, we have 77 true positives and 33 true negatives, with only 1 false positive and 6 false negatives. While directly compared to the Full-Model, we can see a higher representation of negative predictions since the model eliminates 192 rows of data to produce a 2:1 positive to negative lung cancer diagnosis ratio.

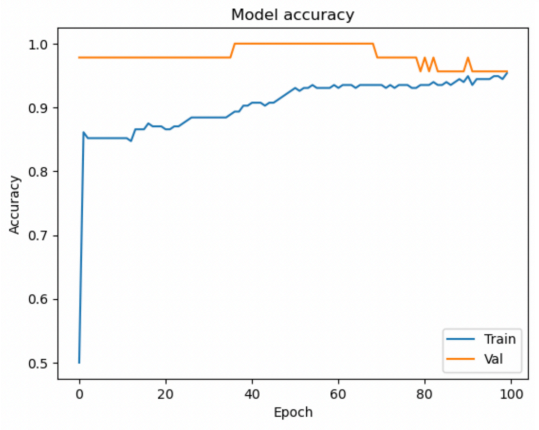
5.3.3 Logistic Regression Unbiased



The Logistic Regression Unbiased model is the least accurate of the three, but still produces satisfactory results at a total of 93.75 percent accuracy. There were a total of 37 true positives and 36 true negatives, with only 2 false negatives and 3 false positives. Overall, all three Logistic Regression models were very effective at predicting lung cancer.

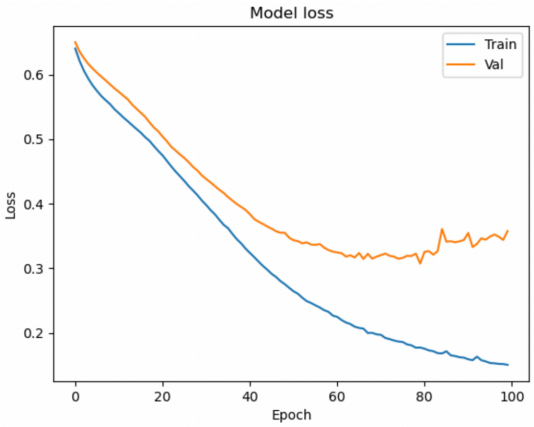
5.3.4 Neural Network Full Model

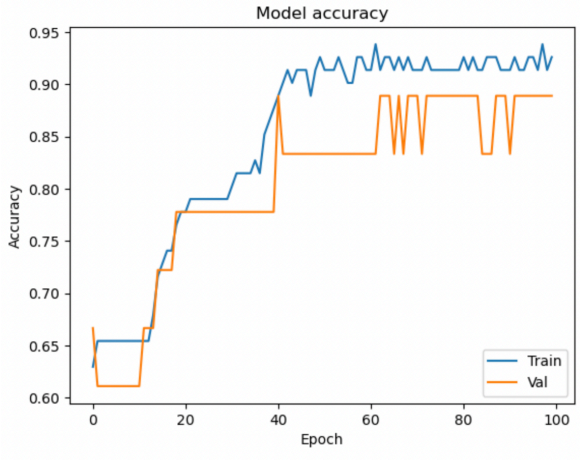




The Neural Network Full Model had moderately well fitted results, with the model loss showing similar curves between the training data and validation data, which both achieved a loss below 20 percent by the end of their epochs. The accuracy curves are not quite as well fitted but both taper to a similar accuracy by the end of their epochs of about 90 percent.

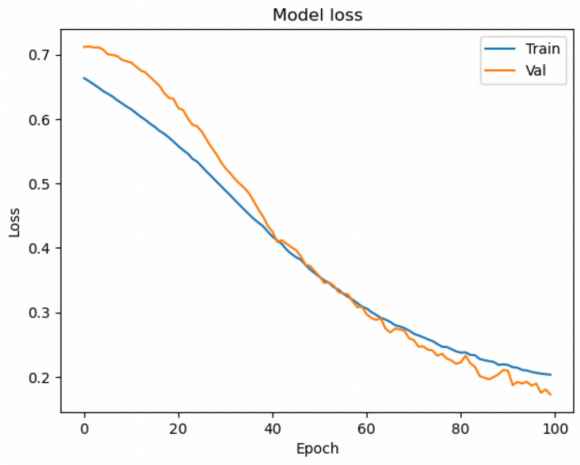
5.3.5 Neural Network Half-Biased

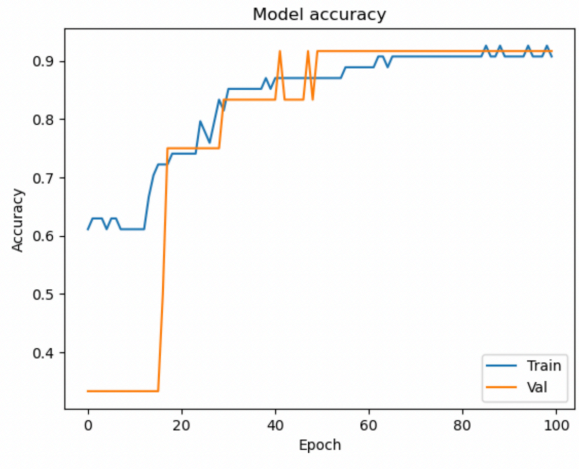




The Neural Network Half-Biased model was decidedly the least fitted model, which is especially clear when analyzing the loss of the model. Both validation and training data curves follow roughly the same paths for the first half of their epochs, but after epoch 40 they diverge significantly with the validation data approaching a los upwards of 40 percent while the training data fell below a 10 percent loss. The accuracy of the model was not quite as bad, but followed a similar trend. The curves relatively follow the same paths up until epoch 40, when the validation data again starts to diverge from the training data. The validation data ends at under 90 percent accuracy while the training data is over 90 percent accuracy. This is still not a good fit, but not quite as bad as the divergence in the model's loss.

5.3.6 Neural Network Unbiased





The Neural Network Unbiased model was definitively the best fitted model. For the model loss, both the training data and validation data follow near identical curves for the entirety of their epochs. For the model accuracy, there is some discrepancy between the training and validation data at the beginning of their epochs with the validation data starting at an accuracy of sub 40 percent and the training data above 60 percent, but just before epoch 20 the two curves meet and follow relatively similar curves for the remainder of their epochs. The loss of the model ended at around 20 percent and the accuracy of the model ended at about 90 percent.

7. Conclusion

In this study, I sought out a machine learning algorithm that could input a series of self-diagnosed symptoms of lung cancer and output whether a patient is positive or negative for lung cancer. In recent years allot of research in machine learning has been conducted in the field of medicine to perform related diagnosis of many diseases, and rightfully so. Both Logistic Regression and Neural Networks in this study were extremely efficient and accurate algorithms at predicting lung cancer in patients from self-diagnosed symptoms. Both algorithms and all of their sub-models performed with an accuracy of above 90 percent. Logistic Regression performed the best with the Full model, suggesting that a larger dataset results in a higher accuracy regardless of any existing bias in the data's dependent values. The Neural Network also resulted in a higher accuracy with the Full-model, but was definitively best fitted when there was no bias in the dataset in its Unbiased model. It is safe to conclude that both the Logistic Regression and Neural Network models would be further improved if the dataset in this study was significantly larger and contained little to no bias from the start.

With lung cancer still on the rise in so many countries, and still taking a burdening toll on the modern world as it remains the number one cause of cancer-related deaths, more research in this field needs to be done. The advent of modern computing technology and continued evolution of machine learning algorithms should be further applied to the field of medicine. The earlier a disease such as lung cancer can be detected, the more lives that will be saved.

8. References

1. Alberg, A. J., & Samet, J. M. (2003). Epidemiology of Lung Cancer. *Chest*, *123*(1), 21s–49s.

2. Islami, F., Torre, L. A., & Jemal, A. (2015). Global trends of lung cancer mortality and smoking prevalence. *Translational lung cancer research*, *4*(4), 327–338. https://doi.org/10.3978/j.issn.2218-6751.2015.08.04

3. Kourou, Konstantina, et al. "Machine Learning Applications in Cancer Prognosis and Prediction." *Computational and Structural Biotechnology Journal*, vol. 13, Nov. 2014, pp. 8–17. *ScienceDirect*, www.sciencedirect.com/science/article/pii/S2001037014000464.

4. Levitsky, A., Pernemalm, M., Bernhardson, BM. *et al.* Early symptoms and sensations as predictors of lung cancer: a machine learning multivariate model. *Sci Rep* **9**, 16504 (2019). https://doi.org/10.1038/s41598-019-52915-x

5. Schober, P., & Vetter, T. R. (2021). Logistic Regression in Medical Research. *Anesthesia and analgesia*, *132*(2), 365–366. https://doi.org/10.1213/ANE.0000000000005247

6. Tu, Jack. "Advantages and Disadvantages of Using Artificial Neural Networks Versus Logistic Regression for Predicting Medical Outcomes." *Journal of Clinical Epidemiology*, vol. 49, no. 11, Sept. 1995, pp. 1225–31. www.sciencedirect.com/science/article/abs/pii/S0895435696000029.

7. Stoltzfus, J.C. (2011), Logistic Regression: A Brief Primer. Academic Emergency Medicine, 18: 1099-1104. https://doi.org/10.1111/j.1553-2712.2011.01185.x

8. Michael A. Nielson, "Neural Networks and Deep Learning", Determination Press, 2015