Tyron Samaroo

People Who Smoke Perhaps Contract Lung Cancer

Background

According to the American Lung Association, lung cancer is the leading cause of cancer deaths in both men and women in the U.S (ALA,2020). It is believed that cigarette smoking is the greatest risk factor contributing to this largely preventable disease. Currently, nearly 90 percent of lung cancer cases are among cigarette smokers.

While there are many other risk factors, smoking is considered the worst. The reason why smoking is the leading cause of lung cancer is because it contains a toxic mix of more than 7,000 chemicals (CDC, 2019). Many of these chemicals were tested and recorded as being poisonous which has a direct correlation to causing cancer in people. There are also facts that show that even if one is not a frequent smoker, they are still at risk of contracting lung cancer.

Modeling Problem

This problem can be expressed into a model. A model is interpreted as being an abstraction or an approximation to reality, absolute truth, system, or phenomenon. There are many different models but a mathematical model may be the most useful as it can help one understand a phenomenon. With mathematics, a relationship can be found between the casual inputs or variables that may or may not be dependent on the phenomenon. Without this relationship it would be impossible to come up with a way to validate whether the model works or not. To put this in perspective, a phenomenon or reality would be either an airplane, roads in a city, or success of a person and the models would be a model airplane, a map, or “early to bed, early to rise, makes a man wealthy, healthy and wise” respectively.

One problem is that not all of these models can be validated. In order for a model to be validated, it needs comparable and measurable information or data that can justify it to a certain extent. If the measured data or information provides valuable information, then the model can be considered feasible but if it is not closely related, the model will need to be rebuilt in a way that it can be measured, preferably mathematically. A model is considered a mathematical model if it has numeric inputs and outputs. A navigation route is a perfect model for modeling routes to a given destination as it can take in many measurable information to provide the best possible route. One can figure out how long it will take them to get to their destination.

Now when looking at the model "early to bed, early to rise, makes a man wealthy, healthy and wise" will have the output of wealth, health, and wisdom which is determined by the inputs or features of bedtime and waketime.

$$\begin{bmatrix} health \\ wealth \\ wisdom \end{bmatrix} = f(bedtime, waketime)$$

This model however is imprecise and needs to have numbers and numerical measurements. To account for this, the inputs and outputs must be converted into numerical measurements. One way to go about this is to have bedtime and wake time measured based on the average of a 24 hour time. Health could be the longevity of a person, wealth can be a person's net worth and wisdom can be the amount of books a person read. Now, a more precise model with numerical measurements can be obtained.

Models would seem amazing, all you have to do is find things that are closely related to the phenomenon you are worried about and it would give you exact insight of that phenomenon. For example many formulas in the real world would help model things like the force of an object or its value.One could assume that once a model for this is obtained and has numerical inputs and outputs one is in pretty good shape to understand the phenomenon, but famous statistician George Box and Draper begged to differ. They claimed that “All models are wrong but some are useful.” (Box and Draper , 1987)

Box and Draper are trying to explain that models really do not do a really good job in truly mapping reality. A model’s job is to provide a way for humans to have a better understanding of a problem that is out of the scope of their knowledge.

Thus, a model is just an approximation of the true reality of things and will never be able to really model reality because it ignores many features of reality that may be the true causal inputs. The problem is that one may be stuck because they can never measure these true casual inputs; one does not have access to them because they are out of the scope. The only thing that one can do is approximate as closely as possible to the true casual inputs. It can be defined in an equation below.

$$y = t(z_1, z_2, z_3, ..., z_t) y = t(z_1, z_2, z_3, ..., z_t)$$

The phenomenon y is the output, response, endpoint or dependent variable. The $(z_1, z_2, z_3, ., z_t)$ are the “true” casual input information and the function t is the “true” relationship between the casual inputs and the output. This is the reality that Box is speaking of, one does not know the $(z_1, z_2, z_3, ., z_t)$ that best defines the phenomenon y, so what can be done instead is to find a way to best approximate each of them.

Now, knowing the true causal inputs may pose as a problem so the next best thing to do is to try to define and collect information that can be approximated or related to the true casual inputs $(z_1, z_2, z_3, ., z_t)$. For example, say one wants to model the reality of the creditworthiness of a person. A model can be made where the y will output if a person is either creditworthy or uncreditworthy. y can be written as either $y \in \{creditworthy, uncreditworthy\}$ or $y \in \{0,1\}$ where y is the output space. The true causal inputs would then be the $z's$ that can be defined as the following.

$$\begin{aligned} &z_1 = has\ enough\ money\ at\ time\ of\ loan\ is\ due \in \{0,1\} \\ &z_2 = unforeseen\ emergency \in \{0,1\} \\ &z_3 = criminal\ intent \end{aligned}$$

Looking at this, it is evident that the $\{z_1, z_2, z_3\}$ are unobservable, not able to be measured or inaccessible. So $\{z_1, z_2, z_3\}$ can be approximated by

$$\begin{aligned} &x_1 = average\ salary \\ &x_2 = previous\ loan\ repayment\ missing\ any\ loan\ payment \in \{0,1\} \\ &x_3 = historical\ criminal\ record\ \in \{no\ crime, infraction,\ misdemeanor,\ felony\} \end{aligned}$$

Now, a problem that was discussed before arises once again; the x_3 is not a numerical value. One method that can be done to deal with this is to encode x_3 by breaking it up into additional features, which will result in an input space increase from three to six. This would result in

$$\begin{aligned} &x_4 = infraction \in \{0,1\} \\ &x_5 = misdemeanor \in \{0,1\} \\ &x_6 = felony \in \{0,1\} \end{aligned}$$

Notice that only three more additional features were added since one can determine one of the categories by a linear combination of the others. For example if $[x_1 \ldots., x_4 = 0, x_5 = 0, x_6 = 0]$ then one knows that means no crime was committed. Now one has a new equation that one might be able to deal with

$$y \approx f(x_1, x_2, x_3, ..., x_p)$$
$$y = f(x_1, x_2, x_3, ..., x_p) + \delta$$
$$\delta = t(\vec{z}) - f(\vec{x})\ which\ is\ error\ due\ to\ ignorance$$

Since the $x's$ are what approximate the $z's$ it can be assumed that some function f with some errors due to ignorance, can retrieve the output y. One unique thing to notice is that these inputs for the $x's$ aren't just from ordinary analytical equations like $y = x^2 + 2$ it is an empirical solution which uses data to learn from data.

Supervised Learning

This idea of learning from data is given special meaning especially data that is labeled. When one has historical examples of records and their responses, this is termed supervised learning. Supervised learning has three different ingredients. The first is training data; each of the data points have their corresponding input and output which are all part of the data which can be written as

$$\mathcal{D} = \{< \vec{x_1}, y_1 >, < \vec{x_2}, y_2 >, < \vec{x_2}, y_2 >\}$$

Here $\vec{x_1}$ would be one of the customer characteristics or features and y_1 would be the same exact customer output whether the loan was paid back or not. This follows the same pattern as each labeled data up to x_n characteristic will each have a corresponding y_n response since there are only up to n examples. The next ingredients would the hypothesis set denoted by $\mathcal{H}$. It contains all candidate functions that can approximate f denoted by h. Here again one is making more approximation because there is still so much that they may not know, f is a very complicated function that one can never learn. So what one could do instead is choose a very large set of functions that one thinks will best approximate the f that they are looking for.

$$\mathcal{H} = \{all\ candidate\ functions\ h\ that\ can\ approximate\ f\}$$

Now the last ingredient that is needed to wrap all of this together is an algorithm that will be called $\mathcal{A}$ which uses data $\mathcal{D}$ and $\mathcal{H}$ the hypothesis set and select the best possible function h^* but due to random chance one ends up picking something else that is close enough which will denote this as the g the best candidate function. This function g can be written as $g = \mathcal{A}(\mathcal{D}, \mathcal{H})$. There are many algorithms that can be used to help pick the function g but that will be discussed later.

Errors

Now that one is getting closer to learning something from the data, here will introduce multiple errors or potential errors that may be encountered. It first began with the phenomenon $y = t(\vec{z})$ which is then said to function $f(x) + \delta$ where δ is error due to ignorance since the true function t and it's true casual inputs$(z_1, z_2, z_3, ., z_t)$ would never be known. Then, an algorithm $\mathcal{A}$ was used to find the best possible function h^* but the algorithm might not always find it so presents the phenomenon

$$y = h^*(\vec{x}) + (f(\vec{x}) - h^*(\vec{x})) + (t(\vec{z}) - f(\vec{x}))$$

Yet another error $f(\vec{x}) - h^*(\vec{x})$ is now introduced which is called misspecification error. This error is presented because it is possible that the right hypothesis set $\mathcal{H}$ is not picked to best approximate f . The last error that may be encountered is the possibility of not picking the best h^* in h . Doing so will result with a function which can be denoted as g that may or may not be the h^* so estimation error is concluded if $g \neq h^*$. So now one finally has a model which they call g that best approximates the phenomenon but it has the potential to be filled with errors such as estimation error, misspecification error and error due to ignorance. This can be written as

$$y = \underbrace{g(x)}_{model} + \underbrace{(h^*(x) - g(x))}_{estimation\ error} + \underbrace{(f(x) - h^*(x))}_{missspecification\ error} + \underbrace{(t(z) - f(x))}_{error\ due\ to\ ignorance}$$

Now that it is known that errors will be encountered, it is essential to find ways to deal with them. All of these errors can be broken down simply in supervised learning as just being the difference between the prediction y which will now referred to as $\hat{y}$ which one can get from the model and each labeled data y_i which one can use to understand how well the model has done. There are also some steps that can be taken to help reduce the errors and improve the performance of the model. If $(t(z) - f(x))$ the error due to ignorance is very large. It can be reduced by adding more features that are relevant to the data. The more data that is collected, the better of an understanding one would have of the phenomenon and be closer to the true function with its true inputs. If $(f(x) - h^*(x))$ the misspecification error is large it can be reduced by expanding the hypothesis set $\mathcal{H}$ to introduce more complex functions. And lastly if $(h^*(x) - g(x))$ the estimation error is large. It can be reduced by increasing the sample size. The more data points one has, the better one is able to find the function that can closely predict the phenomenon with great accuracy.

The Model

Now back to the phenomenon that is of interest. The issue if people who smoke cigarettes are susceptible to lung cancer requires a deeper look to understand why. There is a lot of evidence that points towards cigarette smoking being the sole cause of lung cancer Now that it is understood what a model is and how to go about it, the question of whether the model “people who smoke contract lung cancer” is a valid model that can be explored. For this model the response y would output yes or no based on if smoking caused the lung cancer. This is a special type of response space which is called a binary response. It is called a binary response because there's only two possible outcomes, yes and no. The model will take in a certain number of inputs which is collected from patients and based on these inputs it will give a prediction of either yes or no.

However, another issue with this model ignores that there are true casual inputs $(z_1, z_2, z_3, ., z_t)$ that uses some function t to predict the phenomenon in this case which would be lung cancer. It was even stated by the American Lung Association that “It has been estimated that active smoking is responsible for close to 90 percent of lung cancer cases; radon causes 10 percent, occupational exposures to carcinogens account for approximately 9 to 15 percent and outdoor air pollution 1 to 2 percent. Because of the interactions between exposures, the combined attributable risk for lung cancer can exceed 100 percent” (American Lung Association, 2020). This model only took into consideration one of the possible features that contribute to lung cancer and ignored the other possibilities such as exposure to radon, uranium, arsenic, chromium, nickel, carcinogens, history of chronic obstructive pulmonary disease, and other forms of cancer that may have spread to the lungs. To make this model more intuitive, one would first need to consider these as the true causal inputs $(z_1, z_2, z_3, ., z_t)$ so now a better understanding of the current phenomenon is acquired. But these causal inputs $(z_1, z_2, z_3, ., z_t)$ pose as a problem because they are unobserved and impossible to obtain so one must try to collect information that is related to the $(z_1, z_2, z_3, ., z_t)$. Here are some possible features

$$x_1 = number\ of\ months\ a\ person\ smoked$$
$$x_2 = measure\ air\ quality\ where\ person\ lived$$
$$x_3 = type\ of\ other\ cancer\ person\ have \in \{breast, bone, skin, ...\}$$
$$x_4 = measure\ content\ of\ radon\ in\ person\ blood$$
$$x_5 = measure\ content\ of\ uranium\ in\ person\ blood$$

All of the features are continuous except for x_3 that is a categorical variable that spans more than 200 types of potential cancers. As done before, one can easily deal with this by encoding it by adding more feature, exactly 199 more features to our data $\mathcal{D}$. For instance, if a person has breast cancer there will be a one to denote True or yes for breast cancer and zero to denote False or no for any other type of cancer. All of the data is now easily obtainable and ready to be used for learning. For notation purposes the features x_1, x_2, x_3, x_4, x_5 would be denoted as p. These x_1, x_2, x_3, x_4, x_5 are reasonable for approximating as they are common things that are related to lung cancer for a given patient. $x_1, x_2, x_3, x_4, x_5$are easily recoverable since patient data is always logged when they make a trip to the doctor. CDC, one the major health organizations, makes all this information easily available.

Another problem that might occur is not having enough examples which will be called n. Having a sizable n of about 5,000 to 10,000 would be reasonable to understand lung cancer. Having any more would also be better as more data is collected, and as n increases the more the phenomenon would be understood.

All of the data is also labeled so each feature of a person above is represented in a vector and each feature of a specific person can be referenced by each column. The data can be represented as a $n \times p$ matrix where n denotes the number of persons in the data $\mathcal{D}$ and p being the number of features that is collected denoted by the $x's$.In the words, there are things that cannot be truly predicted unless everything is known. Thus in the current phenomenon, predicting the solution is a problem since it can be stated that unless one knows every single feature that contributes to lung cancer it still remains unknown. Now all there is left to do is to look for some optimal function f that can approximate with some error due to ignorance,the true function t. f that can approximate with some error due to ignorance what features contribute to the phenomenon of lung cancer. f can be any function and it's an unknown function so the next best thing that can be done, is define a hypothesis set $\mathcal{H}$ which could potentially locate f which will then be approximated by a function denoted by h^*. One problem as mentioned before is that if the prediction is wrong there is potential to induce two errors, error due to ignorance from before hand and now misspecification error that will happen if the wrong hypothesis set $\mathcal{H}$ is chosen to search for the h^* that best approximates f. One potential hypothesis set can be represented by

$$\mathcal{H} = \{\mathit{1}_{w_o \vec{x}} > 0 : w \in R^{p+1}\}$$

Here an indicator function will return True or one if the weighted vector $\vec{w} * \vec{x}$ If right, it provides a better understanding of lung cancer in perspective of our own human knowledge and avoids these errors but that is very rare. Looking at the hypothesis set $\mathcal{H}$ for h^*, yet again another problem occurs, no matter what algorithm that is used to find h^* there is a chance that it does not locate it and settle with some other hypothesis that is close enough. The algorithm then return returns a function that will be denoted as g, the best hypothesis that an algorithm $\mathcal{A}$ returns. This also contributes to another error known as estimation error. So one will receive a linear function g that was optimized by the algorithm that was chosen that can best provide a prediction on whether a person will have lung cancer or not. To find g the use of an algorithm known as PLA or better known as Perceptron Learning Algorithm can predict the given phenomenon of lung cancer.

The Perceptron algorithm works by first initializing the weight $\vec{w}$ to $\vec{0}$ or random. After this the first prediction is calculated based on the indicator function $\hat{y}_i = \mathit{1}_{w_o\vec{x}} > 0$ Next one updates all the weights from $j = 1, , p + 1$

This results in

$$\begin{aligned}
&w_1 = w_1 + (y_i - \hat{y})1 \\
&w_2 = w_2 + (y_i - \hat{y})x_{0,1} \\
&\vdots \\
&w_{p+1} = w_{p+1} + (y_i - \hat{y})x_{p+1,1}
\end{aligned}$$

Then one would recalculate $\hat{y}_i = \mathit{1}_{w_o\vec{x}} > 0$ for all $i \in \{1, ..., n\}$. Finally one will then repeat all these steps from updating the prediction $\bar{y}$ and the weight vector $\vec{w}$ until they reach a threshold or have completed the maximum number of iterations. Now that the algorithm $\mathcal{A}$ and the data $\mathcal{D}$ are obtained, one is able to compute g. One important thing to note is that if the data $\mathcal{D}$ is linearly separable then the algorithm will converge, if not one would be in trouble.

To do even better we can define a stronger hypothesis set $\mathcal{H}$ and a greater algorithm $\mathcal{A}$. Instead of $\mathcal{H} = \{\mathit{1}_{w_o\vec{x}} > 0 : w \in R^{p+1}\}$ the hypothesis set would be $\mathcal{H} = \{\mathit{1}_{w_1 x \in [\vec{a},b]} + \mathit{1}_{w_2 x \in [\vec{c},d]} + \mathit{1}_{w_3 x \in [\vec{e},f]} + \mathit{1}_{w_4 x \in [\vec{g},h]} > 0 : w \in R^{p+1}\}$, the response space is still $y \in (0, 1)$ and the algorithm Classification Trees will be used.

Classification Tree is basically like a real tree where it stems into branches based on a given decision. The Classification Tree Algorithm is an algorithm that considers every possible orthogonal-to-axis split $x_j \leq x_j \ where\ j = 1 \ldots p \ and\ i \in 1...n - 1$ and then computes the $SSE_l \ and\ SSE_r$ which are the SSE in the left and right node of a classification tree. Gini is the $Gini\ Impurity$ in the left and right node of a classification tree. Gini is the measurement of the likelihood of an incorrect classification of a new instance. Normally SSE is used as the objective function inorder to measure error, but only works in regression talk. This problem is a classification problem so Gini is the perfect substitute. The gini weight average will be as follows. Then at each leaf assignment the prediction $\hat{y}$ will be the mode of all the $y's$.

$$Gini_w = \frac{n_L Gini_L + n_R Gini_R}{n_L + n_R}$$

$$\begin{aligned}&Where\ Gini_L := \textstyle\sum_{K=1}^{K} \hat{p}(1-\hat{p})\\&For\ Gini_R\ same\ applies\ just\ change\ n_L\ to\ n_R\\&Where\ \hat{p_k} := \textstyle\sum_{i=1}^{n_L} \frac{\mathit{1}_{yi}=C_k}{n_L}\end{aligned}$$

Classification Tree is chosen because we have multiple inputs and trees do a good job of providing a richer hypothesis set. Having a better hypothesis set will lead to having a prediction that is more closely related to the phenomenon.

One issue that has to be dealt with is validation of the model. So using the $\mathcal{D}$ that is collected on patients who were labeled as either having lung cancer or not and consist of features that can help determine the reason . In order to validate a model it is important to check its performance on the data $\mathcal{D}$. To do this, split it into two subsets which can be called $D_{train}\ and\ D_{test}$. There is no best way to split the data but it would make sense to make sure that the training data D_{train} is larger than the testing data D_{test}. Some common splits of the data would be 80/20, 75/25 or 50/50 where D_{train} is larger than the testing data D_{test}. So given the split 80/20, 80 percent would be D_{train} and 20 percent would be D_{test}. Now when validating the model, there are about two ways to go about it, either its in-sample or out-of-sample validation. When speaking of in-sample it is referring to how well the model did on the D_{train}. However if one wants to do validation, one can see that this is not really honest validation as it does not provide better insight how the model will perform on future data that it has not seen. About 80% of the patients will be used to compute a model with the algorithm PLA and Classification Trees which was mentioned before to find the optimal function g. Now if an honest validation is desired, it can actually be done since some of our data was put aside as if it was never seen before. This data D_{test} can be thought as if it's been placed in a lock box and can only be accessed after one has their model. When the model is completed, one can check how well it will do on this D_{test} data which is called out of sample validation. So the model g is making predictions on observations that it has never seen. To measure how well a PLA model does, one can calculate errors it makes by using an error metric RMSE known as the root square mean error that does exactly what it sounds like. It first sums all the squared errors known as SSE, note that it is squared because the errors should not cancel each other out. Then the mean of SSE is computed which is called MSE or mean squared error. After that the square root is taken to obtain RSME otherwise known as the root square mean error which is an honest metric to measure errors the model made. The previous terms can be defined as follows

$$SSE = \sum_{i=1}^{n} (y_i - \hat{y_i})^2$$
$$MSE = \frac{1}{n} SSE$$
$$RSME = \sqrt{MSE}$$

It is known that there are many different errors that the model can make. Now looking at the problem we have at hand, it is trying to classify if a patient has lung cancer or not based on the input features. The problem is that PLA will not do a good job since it returns a continuous response and we need a binary response. This is why classification trees are used instead as a better example. Classification Tree will use Gini Impurity to measure performance, as was already mentioned above.

Now, the function g that is obtained predicts the phenomenon of lung cancer which can be written as

$$y = \underbrace{g(x)}_{model} + \underbrace{(h^*(x) - g(x))}_{estimation\ error} + \underbrace{(f(x) - h^*(x))}_{missspecification\ error} + \underbrace{(t(z) - f(x))}_{error\ due\ to\ ignorance}$$

Now to understand why there might be a large error in the predictor, it can be broken down into three different errors. There might have been an issue where PLA did not pick the best line that separates the data so there may have been an overestimate or underestimate. The next problem is that the data is not linearly separable and that another set of functions could have done a better job, thus introducing the wrong specification. It might be a good idea to spend more time on trying to find a set of functions that would best predict well on the data or else there will be huge errors. Now the last error is something that one can do almost nothing about. It is an error due to ignorance meaning that there is a lack of information, so to combat this it will be necessary to collect more information for $\mathcal{D}$. Now if $\mathcal{D}$ contains all the data, one will only get to predict or understand the phenomenon of lung cancer, it then becomes important to acknowledge that this error due to ignorance exists and should be noted for future prediction.

After discussing all these aspects, a model is outputted that can be used to help predict whether a person features x_1, x_2, x_3, x_4 will result in them getting lung cancer or not. When making predictions using a model, precaution is necessary when trying to perform extrapolation. Extrapolation occurs when the features of a person that was not seen before is not in the range of the data $\mathcal{D}$ that was used to build the model. This is very dangerous especially if a new patient inquires about getting lung cancer and is told that there is a high chance that they will not get lung cancer, but they end up actually contracting it. This would cause a lot of trouble, so what should be done instead is use the model to perform interpolation. With interpolation, making predictions are more reliable because it is in the range of the input data $\mathcal{D}$ that was used to generate the model. This way a solid prediction can be offered that can predict the chances a person will contract lung cancer. Thus, this concludes the overview on how to take a phenomenon like lung cancer, find inputs or features x_1, x_2, x_3, x_4 that best estimate the true relationship of lung cancer, and build a honest and valid model in order to learn something useful, in this case whether a person will contract lung cancer.

Bibliography

Box, G. E. P., & Draper, N. R. (1987). *Empirical model-building and response surfaces*. New York: J. Wiley.

Lung Cancer Fact Sheet. (n.d.). Retrieved from https://www.lung.org/lung-health-diseases/lung-disease-lookup/lung-cancer/resource-library/lung-cancer-fact-sheet

Lung Cancer Risk Factors: Smoking & Lung Cancer. (n.d.). Retrieved from https://www.cancer.org/cancer/lung-cancer/causes-risks-prevention/risk-factors.html

Lippincott Williams & Wilkins. (2009). *Pathophysiology made incredibly easy!* Philadelphia.

Park, R. M., Stayner, L. T., Petersen, M. R., Finley-Couch, M., Hornung, R., & Rice, C. (2012, May). Cadmium and lung cancer mortality accounting for simultaneous arsenic exposure. Retrieved from https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4633087/

What Are the Risk Factors for Lung Cancer? (2019, September 18). Retrieved from https://www.cdc.gov/cancer/lung/basic_info/risk_factors.htm