# Capstone Report

## Measuring Performance of Different Statistical Models to Discriminate Against Cancer

The goal of this project is to find a methodology that can best discriminate between participants with and without Non-Hodgkin Lymphoma based on exposure to certain chemicals found in pesticides and herbicides. Chemicals used by participants are often trapped in their house's carpet, and overtime through these chemicals can be released into the air and make contact in some way with said participants. Since carpets act as longterm chemical-repositories exposure of these chemicals can last for a long time. To the best of our knowledge there hasn't been a scientific review of different methodologies to assess the relationship between exposure to the chemicals present in this dataset and Non-Hodgkin Lymphoma. Determining if different modeling techniques perform well at discriminating against cancer given a chemical mixture could be greatly beneficial to the medical community.

Previous work has been done detecting single analytes (chemicals) at a time using a "relationship between the analyte and the disease risk modeled..." (Colt, et.al, 2004). However, work has not been done utilizing chemical mixtures (multiple chemicals at a time), to discriminate between positive and negative cancer results in a case-control study setting.

The analytic dataset is from the study "Comparison of pesticide levels in carpet dust and self-reported pest treatment practices in four US sites" (Colt, et.al, 2004), a Non-Hodgkin Lymphoma case-control study in Detroit MI, the state of Iowa, Los Angeles, CA, and Seattle, WA, four areas covered by the Surveillance, Epidemiology, and End Results (SEER) Program of the National Cancer Institute. Cases and controls were matched on age, sex, race, and area. Participants answered a questionnaire and provided dust samples from their vacuum cleaner, provided they had used their vacuum cleaner in the past year and had owned at least half of their carpets or rugs for at least 5 years. Laboratory measurements of the 30 analytes (ng/g or nanograms) contained data subject to limit of detection, and as such missing values were imputed by assigning a value for each missing observation by selecting a value from the assumed distribution using maximum likelihood parameter estimates (Helsel 1990, Moschandreas et al 2001). This imputation was done 10 times, for this project I am using the first iteration of the imputation. The analyte quantities were $log_{10}$transformed.

# Planning, Staging, and Exploration

In order for this project to succeed it needed to be properly scoped. We needed to establish timelines, progress reports, project organization, and how to measure performance.

## Scheduled Meetings

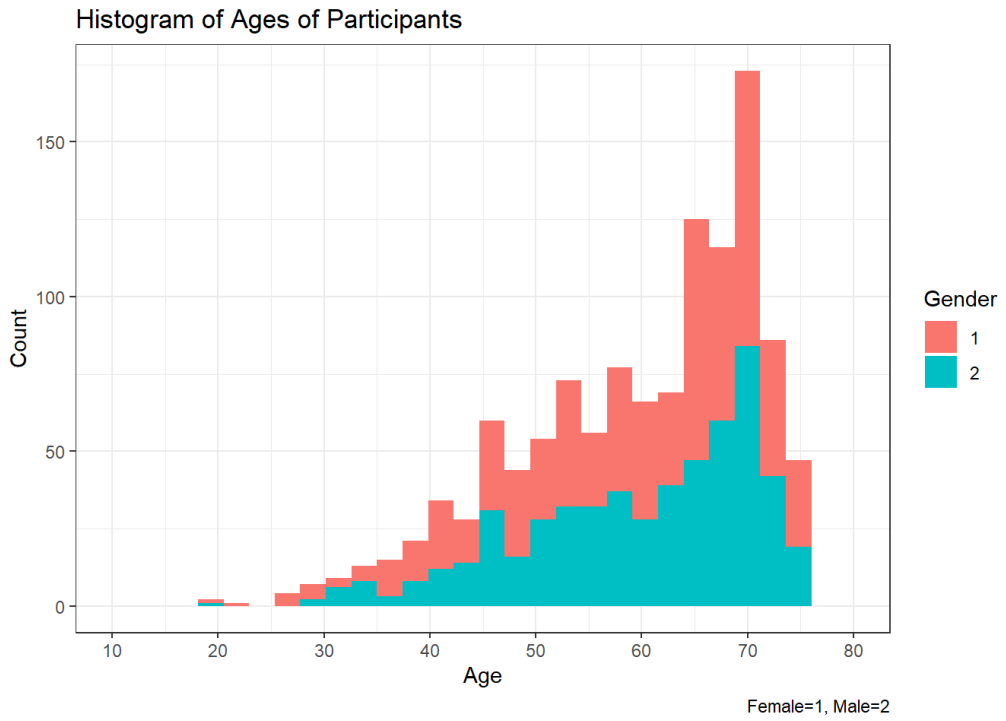
Meetings were held almost a year in advance. Their frequency would increase as time moved forward as a better understanding of what the project would be developed. At first the meetings started with exploring opportunities to eventually monitor project progress.

Meetings became weekly when semester started and work officially began on the project. This would prove to be invaluable as meetings would help resolve issues with development, and help with pivoting to explore other areas of interest such as Risk Prediction.

## Exploration

Work first began by seeing what the data looked like. One of the first things that was observed was how skewed the chemical measurements were. To make the data more normal the chemical measurements were transformed by Log base 10. Histograms of all the chemicals can be found in the chapter "*Capstone: Exploratory Analysis*".

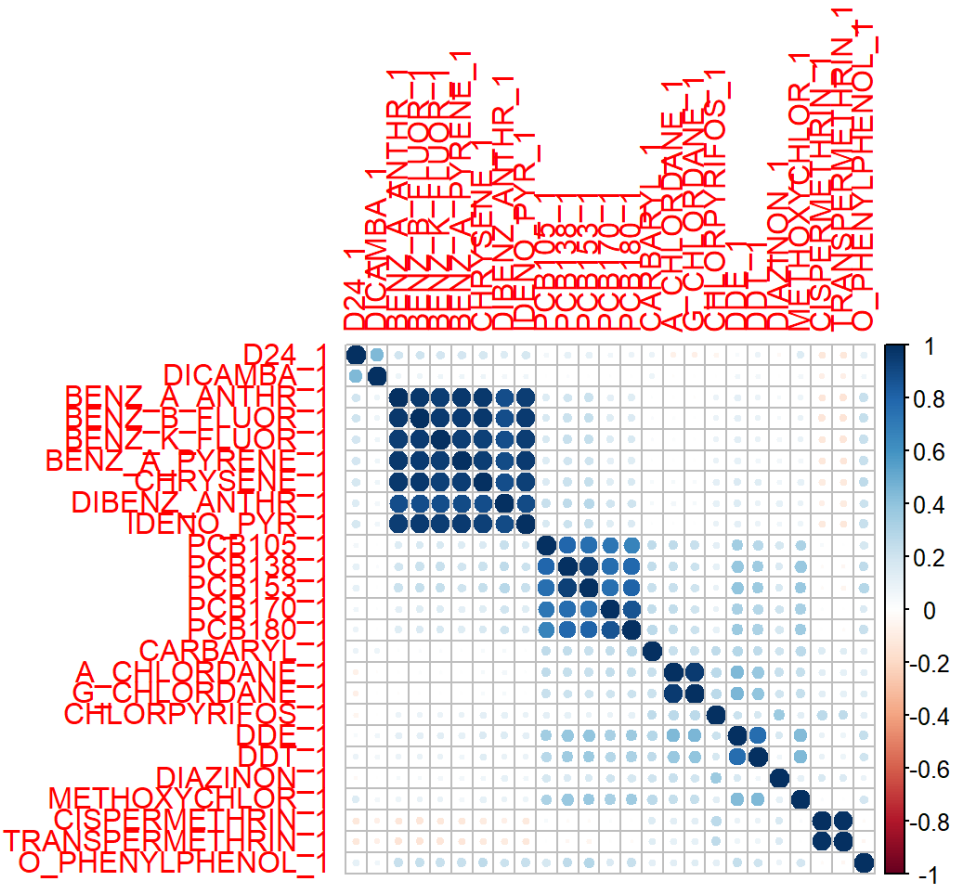
It was also important to consider how other aspects of the data would factor into the discriminating models; these were Sex, Age, and Location of participants.




Above is a histogram of participants' ages and coloring of how many were of what sex. It was found that ages and sex seemed to be well represented across the data, with some skewing towards older participants. It was also determined that participants regardless of age or sex had a relatively even chance of contracting cancer so both variables would be dropped for building the models.

Time was taken to see if there were drastic differences between locations and cases. However the differences appear to simply be in magnitude, but not in proportion of positive or negative cases. So Location was also dropped from the variables used in modeling.

Besides looking at the demographic variables we also took into account the high correlation between variables that would likely exist.




As you can see in the correlation matrix above that was indeed the case. There were distinct groups that seemed to trend together, these are large groupings in dark blue.

## Measures of Performance

In order to determine which models performed better or worse it was decided to measure a model's **Sensitivity, Specificity, PPV,** and **NPV.** Each model would go through 10% Cross Validation and the responses would be compared to the observed values to see how effective it performed discrimination. Each fold would represent 10% of the entire dataset. This works the same way as 10 Fold or Leave One Out, but instead of K number of rows observed this is the proportion of K number of rows taken from the original dataset.

# Models Used and Their Performance

This section describes how each method used for discriminating against cancer worked and performed. The unit of measure for performance used was **Sensitivity, Specificity, PPV,** and **NPV.** Each method utilized 10% Cross Validation to produce results.

# Logistic Regression

## Methods:

Logistic Regression is a commonly used procedure and models the probability of a binary outcome as a function of a series of covariates.

$$P(y^i = 1) = \frac{1}{1 + e^{-(\beta_0 + \beta_1 x_1^i + ... + \beta_p x_p^p)}}$$

The coefficients represent a one unit change in the log odds of the binary outcome associated with a one unit change in the predictors. Model selection was performed by minimizing AIC using stepwise selection. AIC stepwise selection looks to find the simplest model without sacrificing too much on performances. For our stepwise selection we chose “backward”, starting with a full model with all the variables present, and then removing the ones with the highest AIC in the model; this gives a new model and the cycle repeats itself.

## Results:

The model for Logistic Regression, built using Stepwise AIC, became the following:

| | Estimate | Std. Error | z value | Pr(>\|z\|) |
|---|---|---|---|---|
| (Intercept) | 1.3067169 | 0.3950080 | 3.308077 | 0.0009394 |
| **D24_1** | **-0.1632215** | **0.0758236** | **-2.152647** | **0.0313464** |
| **BENZ_K_FLUOR_1** | **0.7551038** | **0.3542921** | **2.131303** | **0.0330642** |
| **BENZ_A_PYRENE_1** | **0.9869700** | **0.3835150** | **2.573485** | **0.0100680** |
| **CHRYSENE_1** | **-1.2036914** | **0.3975767** | **-3.027571** | **0.0024653** |
| **IDENO_PYR_1** | **-0.6784312** | **0.3754626** | **-1.806921** | **0.0707747** |
| PCB138_1 | -0.4276071 | 0.2752332 | -1.553617 | 0.1202757 |
| **PCB153_1** | **0.5548504** | **0.2796977** | **1.983750** | **0.0472837** |
| **DDE_1** | **0.6176043** | **0.1848721** | **3.340711** | **0.0008356** |
| **DDT_1** | **-0.2817503** | **0.1194367** | **-2.358991** | **0.0183247** |
| **DIAZINON_1** | **-0.1778608** | **0.0714132** | **-2.490586** | **0.0127533** |
| PENTACHLOROPHENOL_1 | -0.0000439 | 0.0000297 | -1.480376 | 0.1387730 |

Terms in Blue were found to be significant with P-Values below 0.05, and terms in Green were found to be significant with P-Values less than 0.1, when using a probability threshold of 0.5. Meaning probabilities below 0.5 were considered Negative and probabilities at or above 0.5 were considered Positive.

The following table shows the average error rate from performing Cross Validation:

| Avg. Training Error | Avg. Validation Error |
|---|---|
| 0.3910836 | 0.4135593 |

Below is the confusion matrix and corresponding error rates from performing Cross Validation:

| | Observed NO | Observed YES |
|---|---|---|
| Predicted NO | 131 | 111 |
| Predicted YES | 377 | 561 |

| Error Rate | False Positive Rate (Fall-Out) | False Negative Rate (Miss Rate) | True Positive Rate (Sensitivity) | True Negative Rate (Specificity) | Precision (PPV) | Negative Predictive Value (NPV) |
|---|---|---|---|---|---|---|
| 0.4135593 | 0.742126 | 0.1651786 | 0.8348214 | 0.257874 | 0.598081 | 0.5413223 |

## Conclusion:

Logistic regression has a good Sensitivity, marking 83% of Positive values correctly. Conversely it has bad Specificity, only marking about 25% of Negative values correctly. Its Precision, nearly 60%, indicates that it does a less than stellar job at correctly assigning Positive Values. Likewise its NPV, at 54%, performs similarly at correctly assigning Negative Values.

# LASSO Regression

## Methods:

Least Absolute Shrinkage and Selection Operator (LASSO) is a type of regression that utilizes “shrinkage” to weed out variables that are not optimal for prediction. This is a great option to consider when the number of features is fairly large as it can reduce feature space significantly.

Lasso regression relies on Regularization. Regularization is part of what makes Lasso so special as it helps counteract overfitting data by adding a penalty term, λ. For Lasso it can be written formally as:

$$\sum_{i=1}^{n}(y_i - \sum_{j} x_{ij}\beta_j)^2 + \lambda \sum_{j=1}^{p} |\beta_j|$$

In more layman terms it is “Residual Sum of Squares + λ * (Sum of the absolute value of the magnitude of coefficients)”. To find the best λ cross validation is done across a range of potential values to find the one that makes the model perform the best.

This concept can be applied to Logistic Regression; a binary response which is what we are interested in. The difference between a logistic model and a linear model, is that the Sum of the Likelihoods are optimized instead of the Squared Residuals. This takes the form of the sum of the likelihoods+$\lambda * |slope|$.

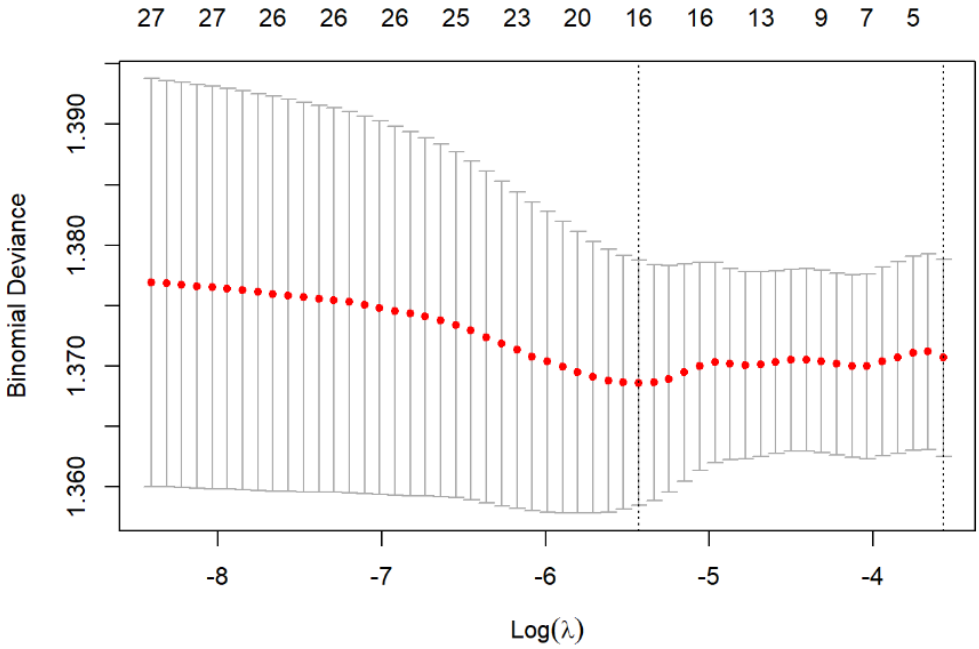
What applying absolute values to the coefficients for penalties allows is having the slope be zero, *instead of asymptotically like Ridge.* Therefore this will eliminate terms/features that are considered “useless”. This makes Lasso **slightly better** than Ridge at *reducing the variance in the model that contains a lot of impractical variables*. Conversely Ridge does **slightly better** than Lasso *when most variables are considered useful.* For this project Lasso was chosen since there are a lot of variables in this data set and reducing the number of them would make interpretation much easier.

## Results:

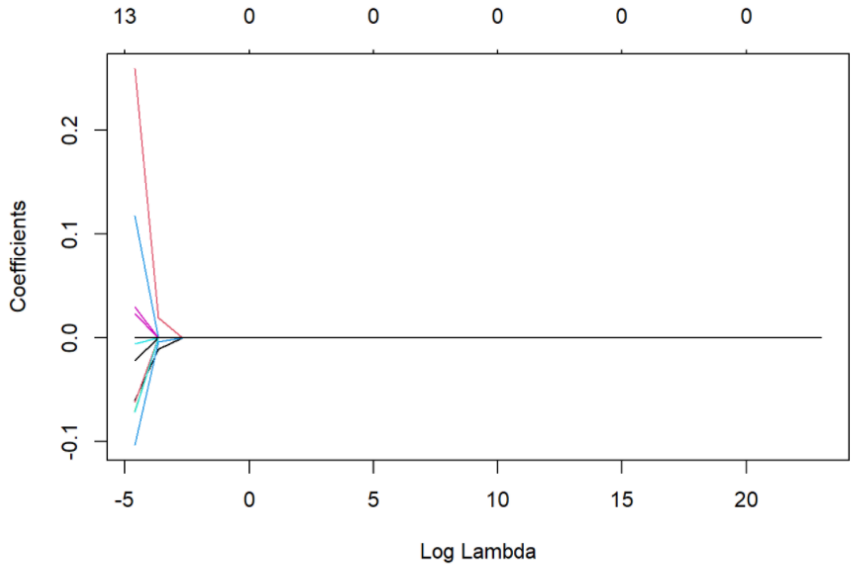
Below is the model built using LASSO Regression, using the best λ(0.004377151)

| | Coefficient |
|---|---|
| (Intercept) | 0.4474444 |
| D24_1 | -0.0597175 |
| DICAMBA_1 | -0.0615908 |
| BENZ_A_PYRENE_1 | 0.0005441 |
| PCB180_1 | 0.1175047 |
| CARBARYL_1 | -0.0054588 |
| A_CHLORDANE_1 | 0.0296104 |
| CHLORPYRIFOS_1 | -0.0219551 |
| DDE_1 | 0.2590521 |
| DDT_1 | -0.0698969 |
| DIAZINON_1 | -0.1031587 |
| METHOXYCHLOR_1 | -0.0715813 |
| PROPOXUR_1 | 0.0230676 |
| PENTACHLOROPHENOL_1 | -0.0000297 |

We can see many of the variables have been eliminated from this process, some of them being the same variables in the Logistic Regression model.




Above in the Binomial Deviance Vs. Log(λ) the λthat minimizes the cross validated error can be seen as the first dotted-line is 0.004377151, which applying log is -5.431357. The second dotted-line is the 1st standard error from min λ at 0.02813665, and applying log is -3.570682.




The Coefficients Vs. Log(λ)shows that many of the coefficients reach 0 very quickly as lambda increases from -5. When plugging in the best λ into the Log function we get a value of -5.431357, this right before most of the coefficients equal zero on the graph.

The following table shows the average error rate from performing Cross Validation:

| Avg. Training Error | Avg. Validation Error |
|---|---|
| 0.4042952 | 0.4279661 |

Below is the confusion matrix and corresponding error rates from performing Cross Validation:

| | Observed NO | Observed YES |
|---|---|---|
| Predicted NO | 85 | 82 |
| Predicted YES | 423 | 590 |

| Error Rate | False Positive Rate (Fall-Out) | False Negative Rate (Miss Rate) | True Positive Rate (Sensitivity) | True Negative Rate (Specificity) | Precision (PPV) | Negative Predictive Value (NPV) |
|---|---|---|---|---|---|---|
| 0.4279661 | 0.8326772 | 0.1220238 | 0.8779762 | 0.1673228 | 0.5824284 | 0.508982 |

## Conclusion:

LASSO has a good Sensitivity, marking 88% of Positive values correctly. However, it has terrible Specificity, only marking about 17% of Negative values correctly. Its Precision, at 58%, indicates that it does a relatively similar job as Logistic Regression at correctly assigning Positive Values. Its NPV, at 51%, performs barely better than a coin-flip at correctly assigning Negative Values.

# LDA

## Methods:

Linear Discriminatory Analysis (LDA) focuses on maximizing the separability among known categories, in this case it's positive and negative cancer results. This is done via a "cut", or plain, to create the separation. LDA starts by creating a new axis that satisfies two criteria:

1. Maximize the distance between means of each category.
2. Minimize the variation, or "scatter",($s^2$) within each category

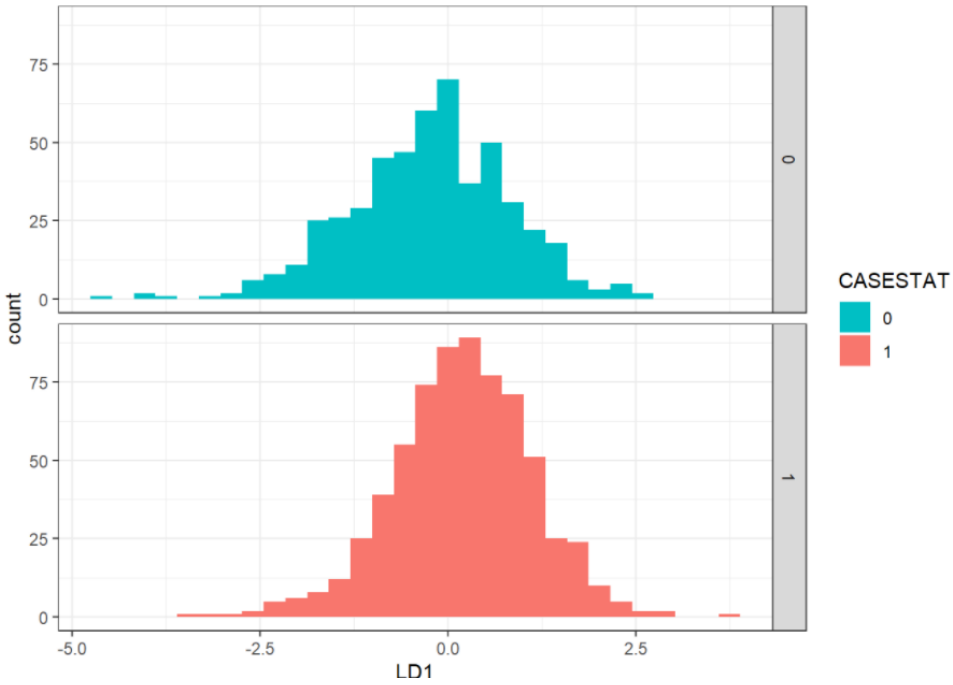
This can be represented as $\frac{(\mu_1 - \mu_2)^2}{s_1^2 + s_2^2}$.A large numerator and a small denominator is ideal. Once this new axis is created points are then projected onto it. This is done on each feature, where "the linear combination of predictor variables that are used to form the LDA decision rule." In other words these variables are used to find the best linear combination of them to help best distinguish data across the two groups. For each individual LDA finds the probability of belonging to the different groups, and whichever probability is the highest per group that individual is assigned to said group.

An important quirk about LDA is that it only works on continuous independent variables because a categorical variable has no mean or variation, it's either one thing or another. This is perfect for the amount of chemicals detected as they are all continuous features.

It functions similarly to Principle Component Analysis (PCA) by creating new axes and reducing dimensions. PCA does this by looking at the variation of features while LDA looks at variation between categories and maximizes their separation.

## Results:

Below is the plot of LD1 from the LDA model




We can see that there is almost a complete cross over between negative and positive cases.

The following table shows the average error rate from performing Cross Validation:

| Avg. Training Error | Avg. Validation Error |
|---|---|
| 0.3841518 | 0.4186441 |

Below is the confusion matrix and corresponding error rates from performing Cross Validation:

| | Observed NO | Observed YES |
|---|---|---|
| Predicted NO | 150 | 136 |
| Predicted YES | 358 | 536 |

| Error Rate | False Positive Rate (Fall-Out) | False Negative Rate (Miss Rate) | True Positive Rate (Sensitivity) | True Negative Rate (Specificity) | Precision (PPV) | Negative Predictive Value (NPV) |
|---|---|---|---|---|---|---|
| 0.4186441 | 0.7047244 | 0.202381 | 0.797619 | 0.2952756 | 0.5995526 | 0.5244755 |

## Conclusion:

LDA has a somewhat good Sensitivity, marking 80% of Positive values correctly. It has bad Specificity, marking about 30% of Negative values correctly, but this is the highest out of the two other models observed so far. Its Precision, at 60%, indicates that it performs similarly to Logistic Regression at assigning Positive Values. Its NPV, at 52%, is about 1% better than Lasso at correctly assigning Negative Values.

# Conditional Decision Tree

## Methods:

**NOTE:** This explanation will assume that the reader has a working understanding of how tree data structures work.

Decision trees work similarly to binary search trees; each node determines how to travel the tree based on data being processed through it. The data can be binary, categorical, or numeric. Each node in the tree is chosen by what node has the lowest impurity (Most common usage is *Gini*). The root node is determined this way, and child nodes are chosen the same way recursively.

Left nodes/leaves are considered *True*, and Right nodes/leaves are considered *True*.

### Steps to build a Tree:

1.

Build a root node. Go through all the variables (temporary root nodes) used to predict an outcome, and then build leaf nodes for each outcome and what that variable's value was (Similar to a confusion matrix). You then go through each temporary root node.

2.

Now with all the candidate root nodes, with their corresponding leaf nodes, are made, we need to calculate their "Impurity"; typically using Gini. Each Leaf node's impurity is calculated as:

$$Impurity = 1 - P(Yes)^2 - P(No)^2$$

**Remember:** The *Left leaf node* is when the **Dependent Variable is TRUE**, and The *Right leaf node* is when the **Dependent Variable is FALSE**. The "Yes" and "No" in the formula above is for the hypothetical Independent Variable and the values it had with the Dependent Variable when it was found to be True or False.

Now that both leaf nodes have their impurity calculated we use them to calculate the overall impurity of the Independent Variable of separating the Dependent Variable. The overall impurity for the current internal node is the weighted average of the leaf node impurities.

It's important to note that the leaf nodes are likely impure; each has a mixture of true and false values of the dependent variable.

$$Impurity_{Overall} = (\frac{(Pop_{Left})}{(Pop_{Left} + Pop_{Right})} * Impurity_{Left}) + (\frac{(Pop_{Right})}{(Pop_{Left} + Pop_{Right})} * Impurity_{Right})$$

3.

After Calculating the Overall Impurity for each Independent Variable we pick the one with the lowest Impurity to be the root of the tree.

4.

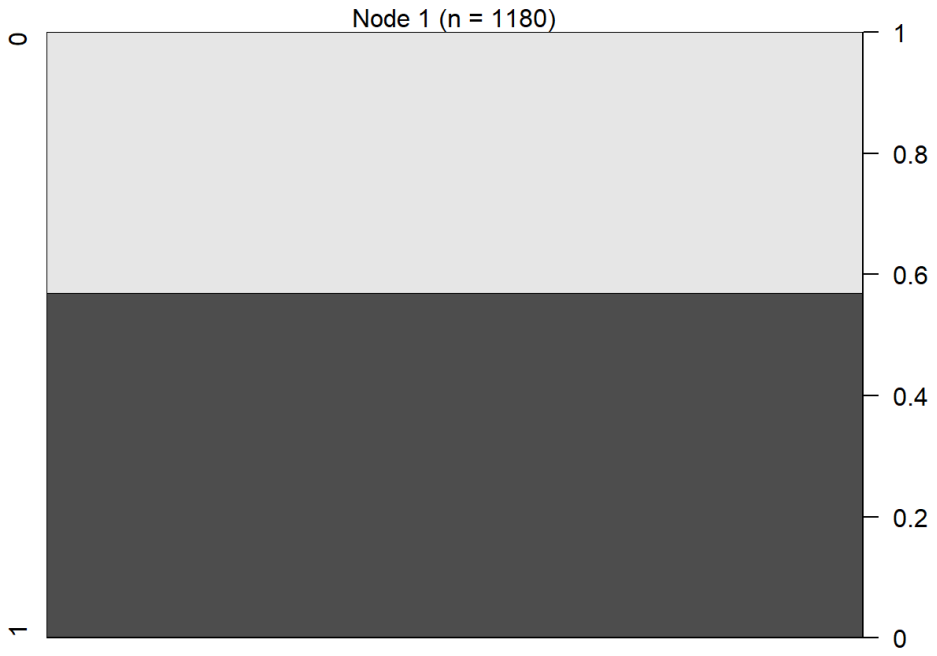
Once we pick the root node we still have to use the other Independent Variables. Starting on the leaf node on the left, we see how well the Population used for that node does for the other Independent Variables. We do the same thing we did before and try to find the Gini Impurity value of each remaining Independent Variable (this is where things start to become recursive).

Once we have the best Independent Variable in terms of the root's left leaf node, now the newest internal node we rinse and repeat until we exhaust all sides of the tree and/or variables that have good impurity values.

Now if a leaf node has a better impurity score than when trying to replace it with an independent variable then we leave it as is. We check the impurity score of the current leaf node, compare it to the candidate variable(s) and if none perform better then that's it.

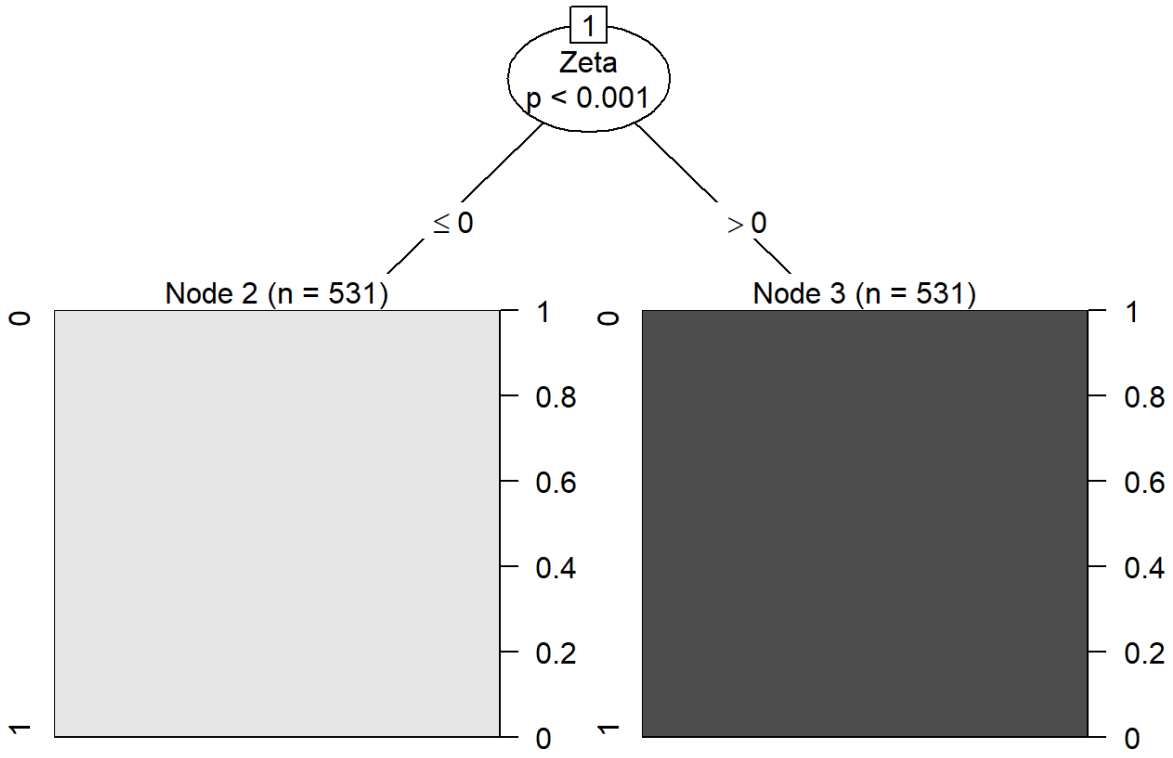
## Results:

Below is the Conditional Decision Tree built.



## Conclusion:

From just looking at the plot above I can safely say this method is worthless to discriminate on cancer status using analyte data alone, as the tree is just the root node. To demonstrate this fact I produced a Conditional Decision Tree with just dummy data below using the same techniques I used to produce the model shown above.




As you can see we actually have a tree-like structure with a root and a few leaf nodes. The root node for the first model I made just shows the average number of positive and negative cases. So all the other features did such a bad job that guessing just based on the average number of positive cases was better at predicting cases; in essence a just slightly better than a coin-flip. For that reason no further effort to measure its performance will be seriously considered.

# Random Forest:

## Methods:

Decision Trees separate different features of data at nodes for classification (if “red” take left path, if “blue" take right path). A Tree usually consists of several branches where each leads to “another tree”; these are recursive structures that contain the same pattern repeated until some edge case in the form of a leaf node.

Random Forests take Decision Trees and make a “forest” of them. Each individual tree in the forest returns a classification result (A or B) and the result with the most votes is what the forest predicts classification as. For example if a forest of 10 trees is used to classify something as *Red* or *Blue*, and 7 trees returned *Blue* then the forest would return *Blue*. The results get democratized!

Random Forests work when a “large number of relatively uncorrelated models (trees) operating as a committee will outperform any of the individual constituent models.” The most important part of a forest working well is that each tree has low correlation between other trees. You can think of each tree as a juror, you want a diverse set of people as jurors because together as a collection should cancel out any other juror’s bias; the last thing you want in a fair trial is a jury that is rigged!

For a good performing tree it needs to follow two criteria:

> 1: There needs to be some actual signal in our features so that models built using those features do better than random guessing.
>
> 2: The predictions (and therefore the errors) made by the individual trees need to have low correlations with each other.

In order to make a forest diverse, in other words making sure each tree has low correlation with each other, we can use two well used options: **Bagging (Bootstrap Aggregation)** or **Feature Randomness**.

- Bagging: Each tree samples with replacement.
- Feature Randomness: Some trees get some features and others don’t. (this is the technique chosen for this project)

## Results:

The following table shows the average error rate from performing Cross Validation:

| Avg. Training Error | Avg. Validation Error |
|---|---|
| 0.4387513 | 0.4483051 |

Below is the confusion matrix and corresponding error rates from performing Cross Validation:

| | Observed NO | Observed YES |
|---|---|---|
| Predicted NO | 147 | 168 |
| Predicted YES | 361 | 504 |

| Error Rate | False Positive Rate (Fall-Out) | False Negative Rate (Miss Rate) | True Positive Rate (Sensitivity) | True Negative Rate (Specificity) | Precision (PPV) | Negative Predictive Value (NPV) |
|---|---|---|---|---|---|---|
| 0.4483051 | 0.7106299 | 0.25 | 0.75 | 0.2893701 | 0.582659 | 0.4666667 |

## Conclusion:

Random Forest has a somewhat decent Sensitivity, marking 75% of Positive values correctly, granted this is the worst performer by far. It also has bad Specificity, marking about 29% of Negative values correctly, but this is a close second to LDA. Its Precision, at 58%, indicates that it performs similarly to other models at assigning Positive Values. Its NPV, at 47%, is the worst at correctly assigning Negative Values.

# Risk Prediction

Over the course of the project it became clear that the models were not doing a good enough job of discriminating between positive and negative cancer results. So we decided to explore a few of the models in the lens of Public Health.

*Risk Prediction* is a side of stats that really only appears in public health related fields For example a public health question could be:

- **Question Regarding Risk**: How many public housing residents will have a heart attack this year?
- **Decision Based upon Risk**: Quantity and location of defibrillators in housing projects.

Another example, in terms of Clinical Medicine could be:

- **Question Regarding Risk**: Will I have a migraine?
- **Decision Based upon Risk**: painkillers or no painkillers

To develop a meaningful Risk Prediction model it needs to follow the procedure called "Transparent Reporting of a multivariable prediction model for Individual Prognosis or Diagnosis", or just TRIPOD for short. TRIPOD has three steps: Calibration, Discrimination, and Decision Analysis.

## Calibration

Here we're looking for "overall performance". If you're using a logistic regression model you can do a goodness of fit test like Hosmer Lemeshow. The Hosmer-Lemeshow test will provide a p-value, which reflects the probability that your null hypothesis – that there is no difference between the distribution of predicted and observed outcomes across quantiles – is correct (in other words, this is a situation in which a very large p-value should make the analyst quite pleased). Often people use Decile Plots, in conjunction with the Hosmer Lemeshow test using Calibration Plots with predicted values on the x-axis and observed values on the y-axis are used to see how well your model makes predictions. In it you'll want to see a slope close to 1, less than 1 indicates overfitted models and greater than 1 indicates under fitted models.

## Discrimination

Now that we've calibrated the model it's time to see how well a model differentiates between subjects with a certain outcome. What is the *probability* that you don't have the disease/condition given a negative test?

Instead of using predictive values most people trying to develop accurate models rely on "receiver operating curves", or ROCs (you can find these graphs throughout this project). ROC graphs plot *sensitivity* and *specificity*. The diagonal in the graph represents "anticipated performance of a useless test, or chance alone". The best performance for discrimination is

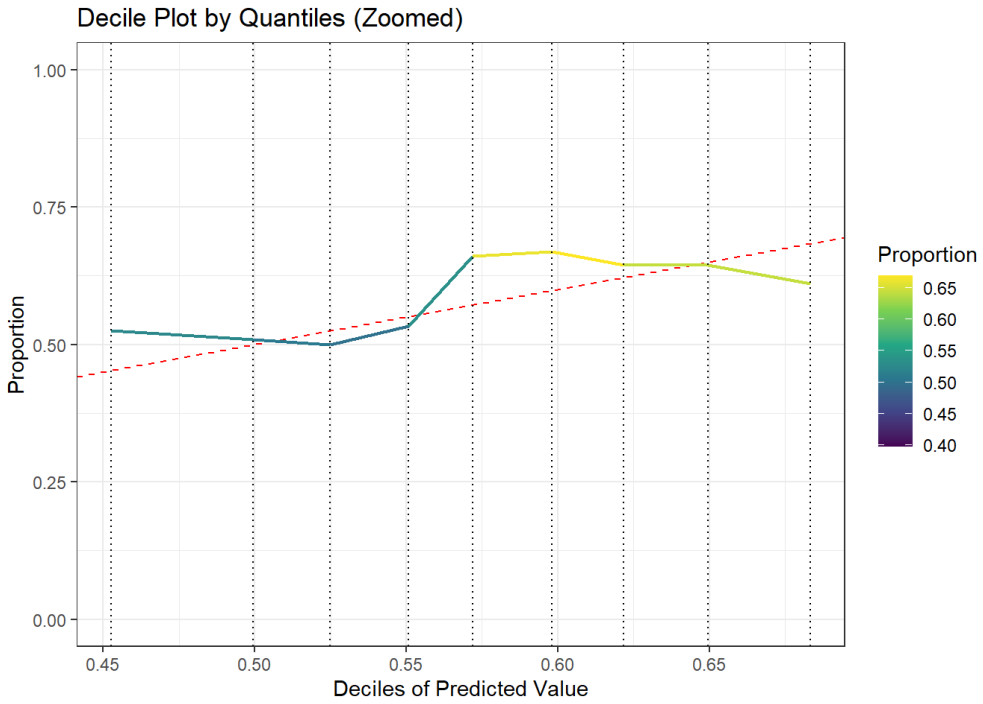
when the line of TPR and FPR gets closer to upper left hand corner of the graph. The greater curve to that upper left corner the greater the *Area Under the Curve* (AUC), which can be used as a stat for performance.

## Decision Analysis

This is the last step and it comes down to simply how practical of a solution we can come up with, or “how and when will the predictions impact actual decisions?” This is where the “rubber meets the road” when trying to make a solution practical or even trying a solution. For example given a probability for having a minor disease would it be worth it to conduct an invasive biopsy? This part was not explored for this project, but could be considered for future work.

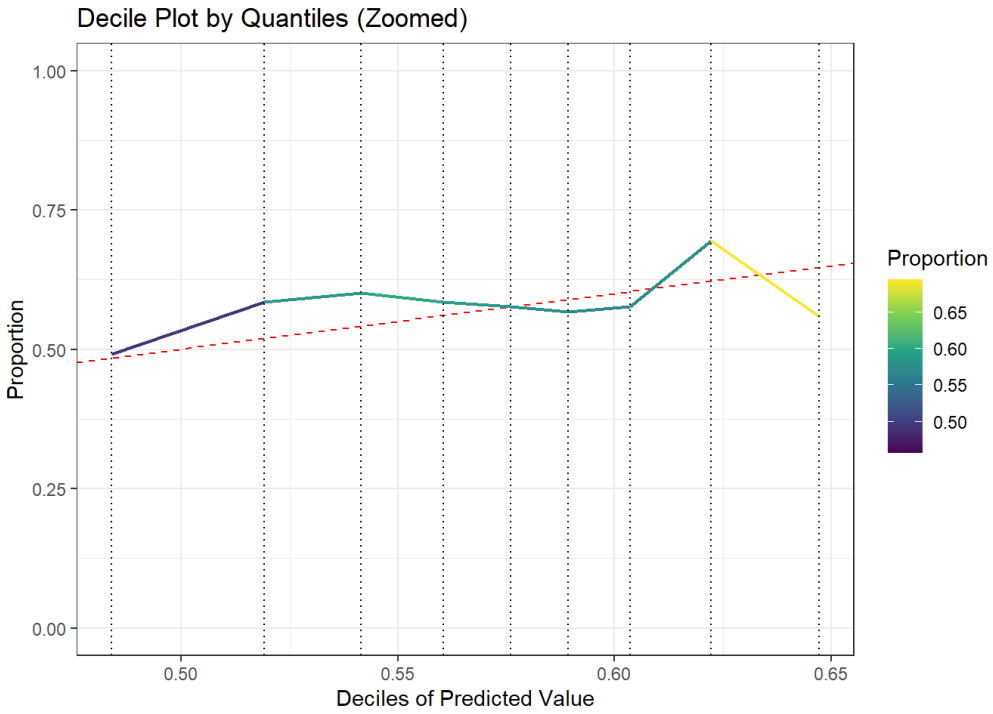
# Logistic Regression

The graph below shows how well calibrated Logistic Regression is across all the quantiles. The range of decile bins are the ranges of quantiles. Each point on the graph is the Proportion of Positive predicted values per decile. A sign of good calibration is if all the points fall onto the red-dotted line (slope being 1). Most of the points fall close to that line within this range of Predicted Values. Only a few Proportion values within the range of Deciles 0.575 and 0.6 are notably higher than the red line. This shows that using quantiles for decile bins is pretty well calibrated, meaning that it could be useful to look at when considering a model for risk prediction.



# LASSO

The graph below shows how well calibrated LASSO is across all the quantiles.The range of decile bins are the ranges of quantiles. Each point on the graph is the Proportion of Positive predicted values per decile. Most of the points fall close to that line within this range of Predicted Values. The only notable problem with this graph is the tail end of the range of deciles, with values spiking above and then below the red line. Overall the points are much tighter around the red line than Logistic Regression. This shows that using quantiles for decile bins LASSO is also pretty well calibrated, rivaling the performance seen in Logistic Regression.



# Review

## Conclusion

| | Error Rate | False Positive Rate (Fall-Out) | False Negative Rate (Miss Rate) | True Positive Rate (Sensitivity) | True Negative Rate (Specificity) | Precision (PPV) | Negative Predictive Value (NPV) |
|---|---|---|---|---|---|---|---|
| Logistic | 0.4135593 | 0.7421260 | 0.1651786 | 0.8348214 | 0.2578740 | 0.5980810 | 0.5413223 |
| Lasso | 0.4279661 | 0.8326772 | 0.1220238 | 0.8779762 | 0.1673228 | 0.5824284 | 0.5089820 |
| LDA | 0.4186441 | 0.7047244 | 0.2023810 | 0.7976190 | 0.2952756 | 0.5995526 | 0.5244755 |
| Random Forest | 0.4483051 | 0.7106299 | 0.2500000 | 0.7500000 | 0.2893701 | 0.5826590 | 0.4666667 |

Above is a breakdown of all the Error Rates. We will be focusing on each model's **Sensitivity, Specificity, PPV,** and **NPV** as stated before. For models that returned the probability of being Positive with cancer (Logistic and Lasso) a threshold of 50% was used to ultimately classify a participant being Positive or Negative with cancer. The other models (LDA and Random Forest) did not apply as they just provided classification.

### Sensitivity

A good Sensitivity score is close to 1. In order of best to worst is:

1. **Lasso (0.8780)**
2. Logistic (0.8348)
3. LDA (0.7976)
4. Random Forest (0.75)

The range of percentages of people predicted to have cancer actually having cancer is pretty large here. Lasso does well with nearly 88% of Positive values being accurate, with Random Forest only marking ¾ of Positive values correctly. Both Logistic Regression and LDA do a modest job hovering around 80%.

### Specificity

A good Specificity score is close to 1. In order of best to worst is:

1. **LDA (0.2953)**
2. Random Forest (0.2894)
3. Logistic (0.2579)
4. Lasso (0.1673)

Across the board all the models do a terrible job of accurately discriminating who does **NOT** have cancer. LDA and Random Forest do well comparatively by being around 30%, Logistic Regression marking ¼ of Negative values correctly. In the rear Lasso does terribly at only marking Negative values correctly around 17% of the time.

## Precision (PPV)

A good Precision score is close to 1. In order of best to worst is:

1. **LDA (0.5996)**
2. Logistic (0.5981)
3. Random Forest (0.5827)
4. Lasso (0.5824)

All of the models do about the same across the board, ranging from LDA's 60% to Lasso's 58%. Meaning that regardless of model, if a participant was predicted to be Positive then the probability of that being correct is roughly 60%.

## Negative Predictive Value (NPV)

A good NPV score is close to 1. In order of best to worst is:

1. **Logistic (0.5413)**
2. LDA (0.5245)
3. Lasso (0.5090)
4. Random Forest (0.4667)

There's a bit more spread in NPV for all the models explored. Logistic Regression being the best at 54%, LDA only slightly worse at 52% and Lasso right behind at 51%. Random Forest brings down the range by only having a probability of correctly assigning a Negative value to be about 47%.

## Best Overall Performer

"The Best Model" is hard to pick because it is between two candidates: Logistic Regression and LDA. LDA does the best in two categories, Specificity and Precision. Logistic Regression while only being the best performer in NPV is a strong contender in all other categories; never getting less than 3rd place. The difference in performance between the two modeling techniques for discriminating between people with and without cancer is very close.

## Worst Overall Performer

Deciding this model will be somewhat easier; Random Forest. Lasso would be at the bottom if not for it's surprisingly high Sensitivity! Granted Lasso also does have the most abysmal Specificity to the point of almost canceling out that previous praise. Random Forest never gets above 2nd place.

However in defense of Random Forest there could exist a combination of settings for tree generation that were not explored that could improve the model's overall performance, but none

were found during the course of this project. It is also worth noting that unlike Conditional Decision Trees, which could be thought of as a forest of size 1, actually produced a working model.

# Lessons Learned

While a number of models used in this project I've learned and used before, a few were new to me. However, beyond those new models I learned much more about the subject matter involving epidemiology and imputation than I had anticipated.

## Imputation and LOD

In short, through imputation you can "fill in the gaps" of missing data by looking at values of when that data is present. When a value for a certain chemical was "missing" it was because the Limit of Detection (LOD) could not detect chemical concentrations low enough to be measured. These chemicals were likely present but the devices to measure them were not sensitive enough to accurately measure them. Thus imputation was used to create a more likely picture of what the chemical traces were. However, from working on this project I suspect I have learned the pitfalls of overzealous imputation.

There were several versions of the dataset used for this project; the original dataset with missing values and 10 iterations of imputation (this project used the 1st iteration). In the datapaper(Colt, et.al, 2004) the author's mention that different areas, out of the 4 sampled from, used different chemicals; some had more problems with flying insects while others suffered from weeds and so different problems required different chemical mixture solutions. What I suspect is that through imputation some chemicals were unintentionally "boosted" in value. Why would we impute chemical values found only in herbicides if a participant only used pesticides?

## Risk Prediction

This was a side of statistics I had never heard of until working on this project. Risk Prediction is really only talked about in the public health sphere. An example of Risk Prediction in the frame of Public Health can be "How many strokes will there likely be in urban areas?" and then determining how you respond to your findings "Assigning more or less emergency services to urban areas". Granted the project never went into Decision Analysis we did explore the first two stages, Calibration (overall performance) and Discrimination (predictive values). I only looked at two models for this, Logistic Regression and Lasso, since only models that returned probabilities would work.

# Further Work

I believe that a project solely focused on Risk Prediction using this data could yield some fascinating results. This project mainly focused on how well different models were able to discriminate between participants with and without cancer, but the tangential work into Risk Prediction showed promise.

Given the disappointing results of the Neural Network model, it would be of interest to see how well it would perform on a platform with more resources. A large cluster would be needed to evaluate different configurations of Networks and to even have the memory required to do cross validation.

## Acknowledgements

I can not begin to imagine being able to even start on this project without the help of Dr. Ana Maria Ortega Villa and Dr. Paul Albert. I have learned so much from these two and owe so much of my success to them. It was truly a privilege to have them shepherd me through this.

# Appendix

## Risk Prediction

"A risk prediction model is a mathematical equation that uses patient risk factor data to estimate the probability of a patient experiencing a healthcare outcome. Risk prediction models are widely studied in the cardiothoracic surgical literature with most developed using logistic regression."

## Sensitivity

"The percentage of true positives (e.g. 90% sensitivity = 90% of people who have the target disease will test positive)."

## Specificity

"The percentage of true negatives (e.g. 90% specificity = 90% of people who do not have the target disease will test negative)."

## Precision (PPV)

"The probability that subjects with a positive screening test **truly** have the disease."
How worried should you be about a positive result?

## Negative Predictive Value (NPV)

"The probability that subjects with a negative screening test truly **don't** have the disease."
How reassured should you be about a negative result?

## Limit of Detection (LOD)

"Limit of detection, LOD is the lowest concentration that can be measured (detected) with statistical significance by means of a given analytical procedure."

# Sources

Below are links to sources used in this report and throughout the project as a whole:


- https://pubmed.ncbi.nlm.nih.gov/14726946/
- https://pubmed.ncbi.nlm.nih.gov/33778356/
- https://christophm.github.io/interpretable-ml-book/logistic.html
- https://www.youtube.com/watch?v=CqOfi41LfDw&t=228s&ab_channel=StatQuestwithJoshStarmer
- https://www.youtube.com/watch?v=wl1myxrtQHQ&ab_channel=StatQuestwithJoshStarmer
- https://youtu.be/sDv4f4s2SB8?t=1333
- https://www.youtube.com/watch?v=IN2XmBhILt4&t=515s&ab_channel=StatQuestwithJoshStarmer
- https://towardsdatascience.com/understanding-random-forest-58381e0602d2
- https://www.blopig.com/blog/2017/04/a-very-basic-introduction-to-random-forests-using-r/
- https://rpubs.com/awanindra01/ctree#:~:text=Conditional%20Inference%20trees%2C%20also%20referred,maximizes%20an%20information%20measure%20
- https://www.statisticshowto.com/lasso-regression/
- https://www.mygreatlearning.com/blog/understanding-of-lasso-regression/
- https://www.youtube.com/watch?v=Q81RR3yKn30&t=287s&ab_channel=StatQuestwithJoshStarmer
- https://www.youtube.com/watch?v=azXCzI57Yfc&t=641s&ab_channel=StatQuestwithJoshStarmer
- http://www.sthda.com/english/articles/36-classification-methods-essentials/146-discriminant-analysis-essentials-in-r/
- https://www.publichealth.columbia.edu/research/population-health-methods/risk-prediction
- https://en.wikipedia.org/wiki/Charles_Sanders_Peirce