**Breast Cancer Tumor Classification:**

**Model Analysis Research Paper**

Team 7: Harry Cheung, Lauren Joo, Cecilia Park, Arin Wang, Yuan Yuan

BA 305: Decision-Making with Data

Professor Tsoukalas

May 4th, 2022

|  |
| --- |

**Introduction**

Cancer continues to be a leading cause of concern globally with it being responsible for nearly 1 in 6 deaths in the United States.[[1]](#footnote-0) Of all the different types, one of the most prominent is breast cancer. According to the American Cancer Society, “every 1 in 8 women develop invasive breast cancer, and are predicted to have 300,000 new cases predicted for 2022”.[[2]](#footnote-1) However, despite these alarming statistics, the survival rate remains significantly high depending on how early the tumor is detected. Thus, accurately classifying a tumor as invasive or non-invasive (malignant or benign) has a great impact on increasing the survival rate of a patient.

This subject of tumor classification is the focal point of what our team built the research project on. Our goal was not to create a medically accurate classification model that hospitals can use, but rather a trial and error process of finding the optimal model when exploring the relationship between the different variables and diagnoses. Some of the models that will be discussed in the paper include PCA, Logistic Regression, Decision Trees, and more. This report will focus on our team’s analysis from building and testing different models in the python language on breast cancer tumor classification.

The data set used to do the analysis is a Breast Cancer Dataset with patient data provided on Kaggle. The dataset contains the diagnosis of the tumor as benign or malignant and 11 different variables of the tumor. The variables are as follows:

Identifier:

1) ID number: Different data points (patients)

Y-variables:

1) Diagnosis (M = malignant, B = benign): Dependent Variable

X-Variables:

Eleven real-valued features are computed for each tumor cell nucleus (descriptions and definitions of these 11 features can be found in Appendix 1):

1. Nucleus 2. Radius 3. Texture 4. Perimeter 5. Area 6. Smoothness 7. Compactness

8. Concavity 9. Concave Points 10. Symmetry 11. Fractal dimension

**Preprocessing**

***Uniform pre-processing steps conducted for each model prior to analysis***

As a first step, we performed a relatively uniform data preprocessing on the breast cancer classification dataset before running analyses and models. After loading in the necessary packages and actual dataset, the pre-processing started with converting the “diagnosis” column from ordinal (M = Malignant, B = Benign) to numerical (1 = Malignant, 0 = Benign). Subsequently, we stored the changed “diagnosis” column as the y variable. The next step included dropping specific columns that will not be included as X variables. Specifically, the column “ID” was dropped as it was invaluable to our model. The “Diagnosis” column was also dropped from the X variables because it is our classification variable.

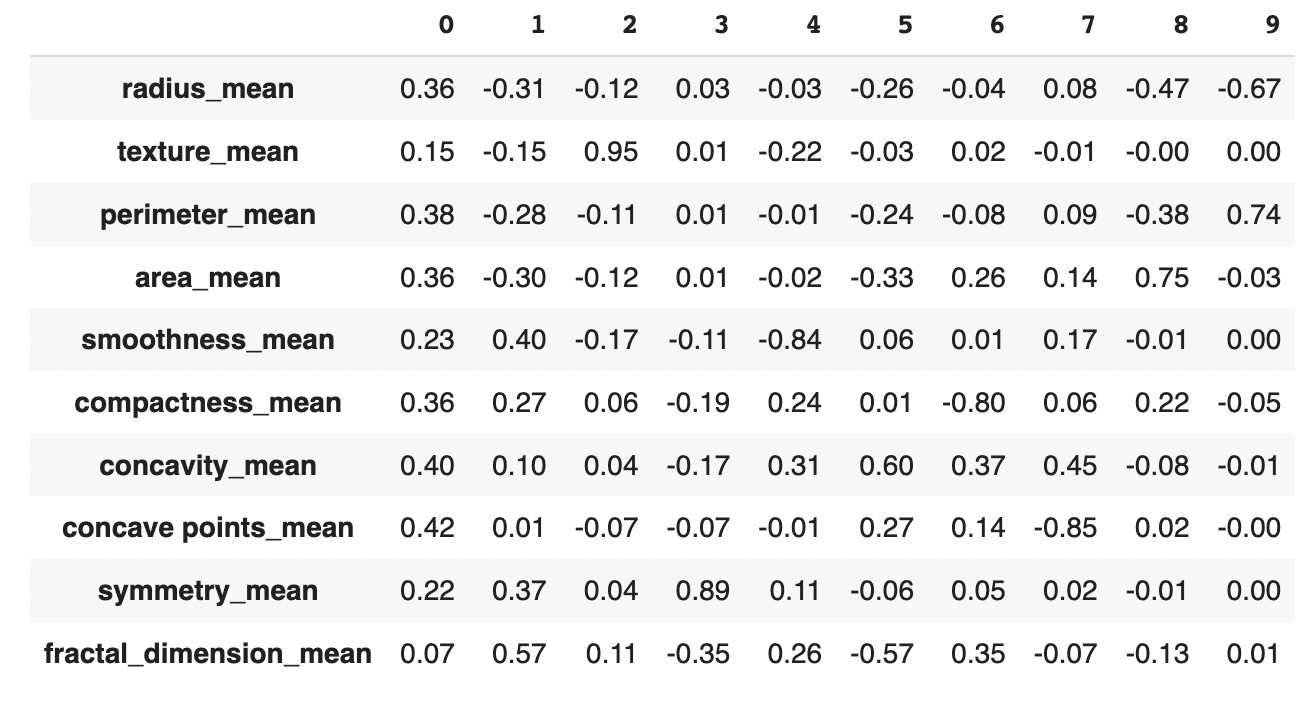
After the more basic steps, we catered to the next lines to make the data fit our research needs better. Our dataset had a vast amount of columns because it contained the mean, worst, and standard error of each variable (i.e perimeter\_mean, perimeter\_worst, and perimeter\_se). Therefore, to be able to separate and run distinct models on each column type, we split the data up into mean, worst, and standard error by utilizing the filter function. We would then drop the unwanted columns to create individual split up data-sets. For example, to create a dataset with only the mean columns, we would drop the standard error and worst columns (df\_se\_col and df\_worst\_col). The final step in pre-processing was to split the data into training and testing for the model with the specifics of test\_size = 0.4, random\_state = 1, and stratify = y.

**Failed Modeling Attempts: PCA & Logistic Regressions**

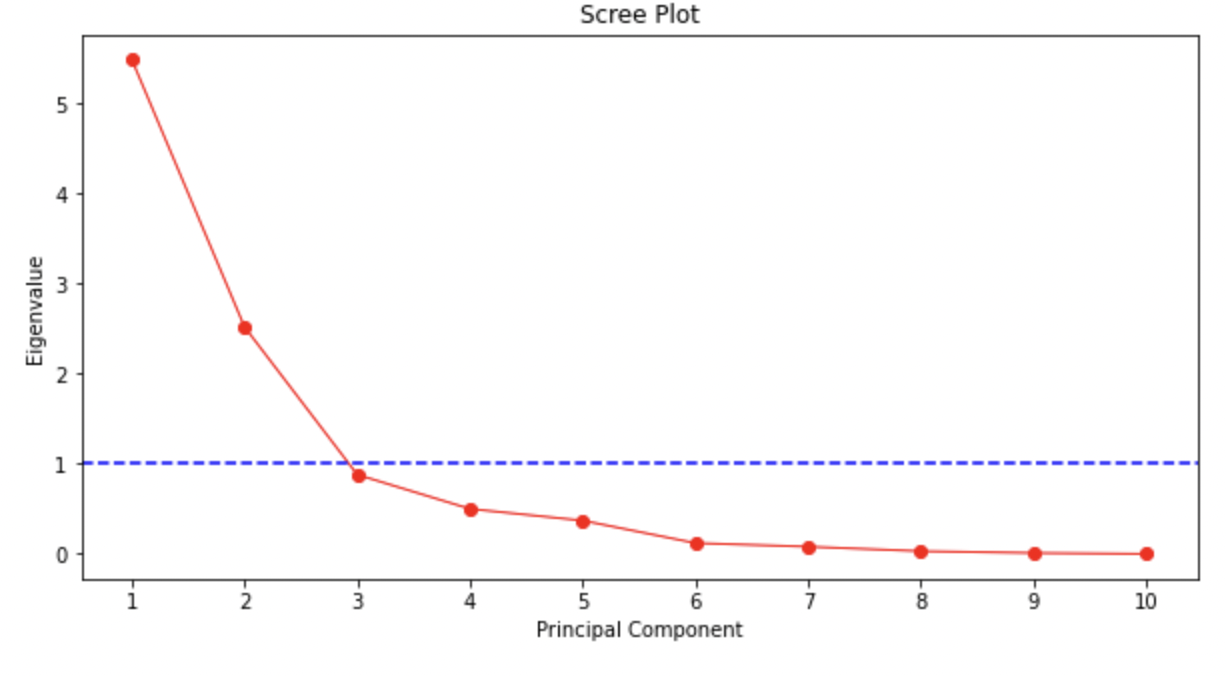
**PCA: Principal Component Analysis**

***PCA does not benefit our analysis because of the inability to interpret principal components***

PCA aims to reduce data dimensions, but does not benefit our analysis. We started with the mean columns and found 10 ambiguous components in our dataset. Then we scaled the dataset and found the percentage of variance explained for each of the 10 principal components. However, this process did not help us define the components. In order to understand the 10 principal components, we generated a table as follows in Figure 1. The 2nd column represents the 1st principal component, the 3rd column represents the 2nd principal component, and so on. The first component is made up of 36% radius\_mean, 15% texture\_mean, etc. It appears to be comprised of random-like ranges of numbers between 0-1 and this pattern is apparent throughout the other components as well. Since there is no correlation between different features, we are unable to interpret what the principal components mean.

[](#D2L_fig_label_Table of Correlation of Principal Components and Different Features)  
Figure 1: Table of Correlation of Principal Components and Different Features

Since we cannot interpret the principal components, we also cannot understand what the two principal components we keep according to the latent root mean (Figure 2). Because our goal is to find the most important features to classify our data and PCA only distracts our analysis, we look to other models like logistic regression.

[](#D2L_fig_label_)  
Figure 2: Eigenvalue Plot for Latent Root Criterion

**Logistic Regressions**

***Logistic regressions are not suitable for our dataset because of the inability to classify data***

Amidst our optimistic exploration into logistic regressions, it failed to capture complicated relationships for our datasets. After fitting the data with all the mean and worst column variables, we get a full logistic regression that includes coefficients for every feature. With the help of this full logistic regression, we extract two features with the largest coefficient and create the reduced logistic regression.

With reduced logistic regression, we test how well the 2 features classify our data by plotting the decision region (Figure 3). The decision boundary is not quite effective, since it cannot clearly separate the two classes. Almost half the data were misclassified, so we then test whether a non-linear model will separate the classes more effectively. We plot with a support vector machine (SVM) with a polynomial kernel to check whether the decision boundary is actually polynomial (Figure 4). Unfortunately, it’s also not suitable since still, nearly half the data suffer from misclassification.

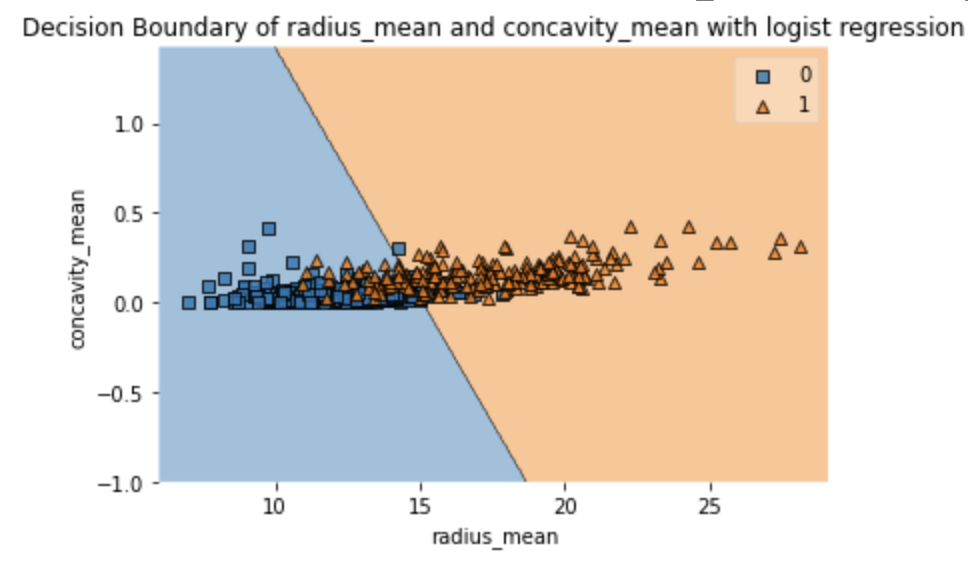
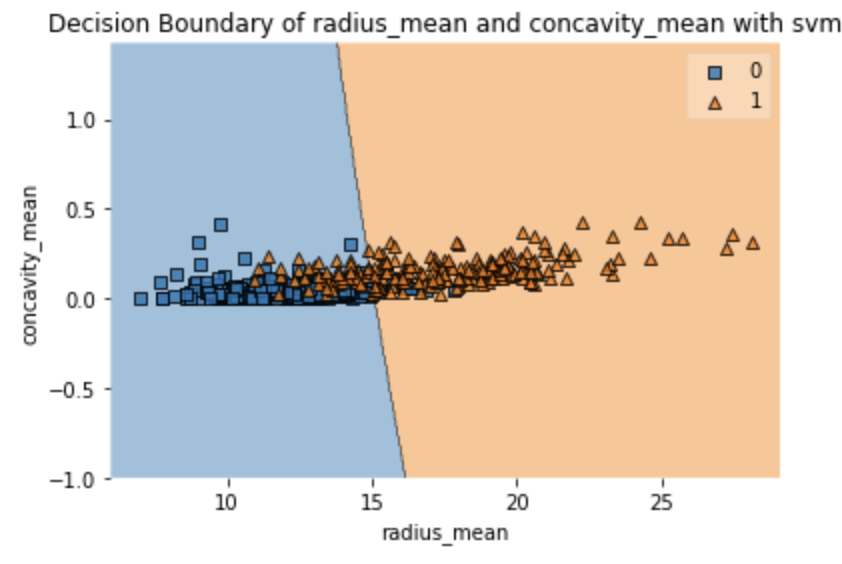
[](#D2L_fig_label_Decision Boundary of radius_mean and concavity_mean with Logistic Regression)[](#D2L_fig_label_)

Figure 3: Decision Boundary of radius\_mean and concavity\_mean with Logistic Regression

Figure 4: Decision Boundary of radius\_mean and concavity\_mean with Support Vector Machine

Next we try a logistic regression for the worst columns, and the 2 features are concavity\_worst and compactness\_worst (Figure 5). However, it is also unable to classify the data. Interestingly, we detect a polynomial part in the decision boundary for the SVM model (Figure 6). Despite this, it is still adequate in clearly separating the two classes.

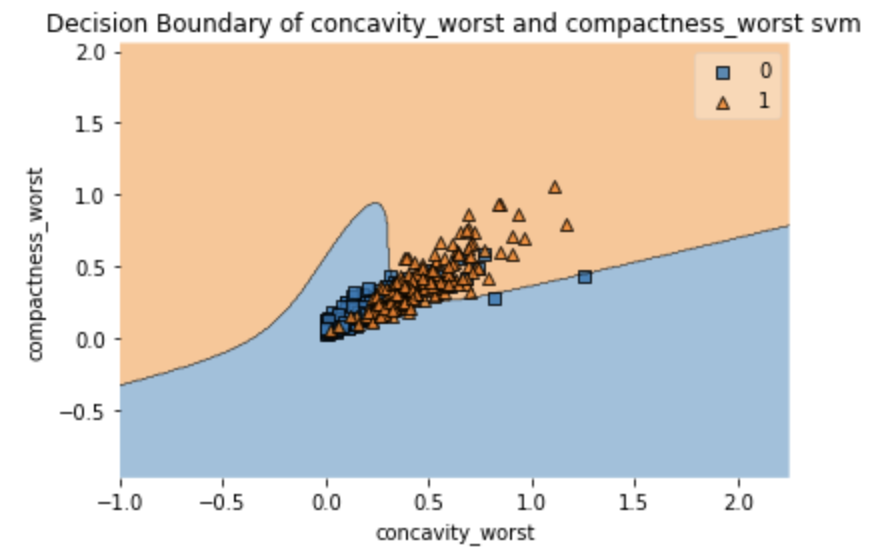
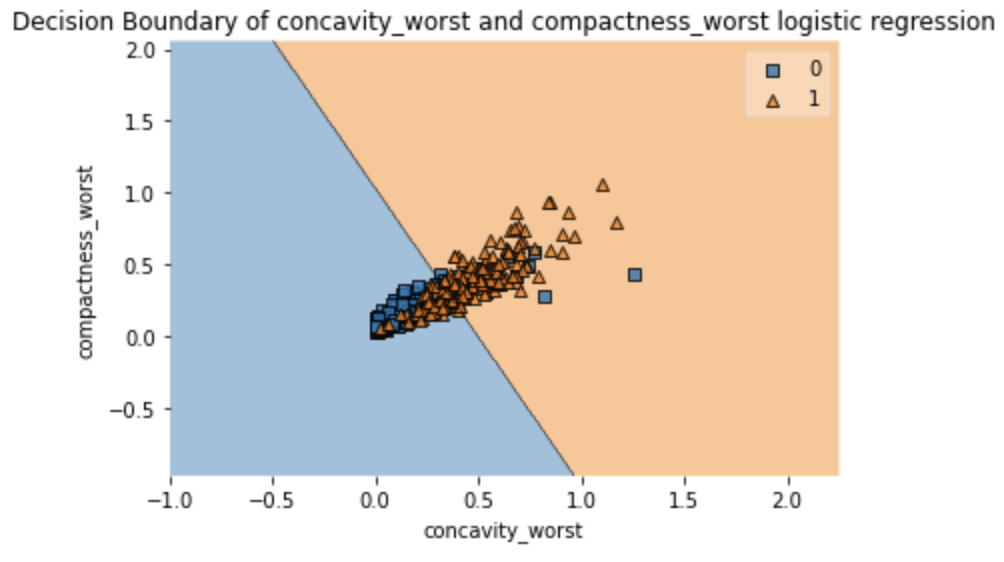
[](#D2L_fig_label_)

Figure 5: Decision Boundary of concavity\_worst and compactness\_worst with Logistic Regression

Figure 6: Decision Boundary of concavity\_worst and compactness\_worst with SVM

Lastly, the analysis is repeated on the two features with the lowest absolute value coefficient. For the mean columns, they are area\_mean and fractal\_dimension\_mean (Figure 7, 8) and perimeter\_worst and area\_worst (Figure 9, 10) for the worst columns. As indicated on the plot graphs, their performances still remained substandard as the others. Like PCA, the logistic models were also not suitable for our dataset. However, we look to other approaches such as splitting our data into 3 separate datasets as countermeasures to our failures.

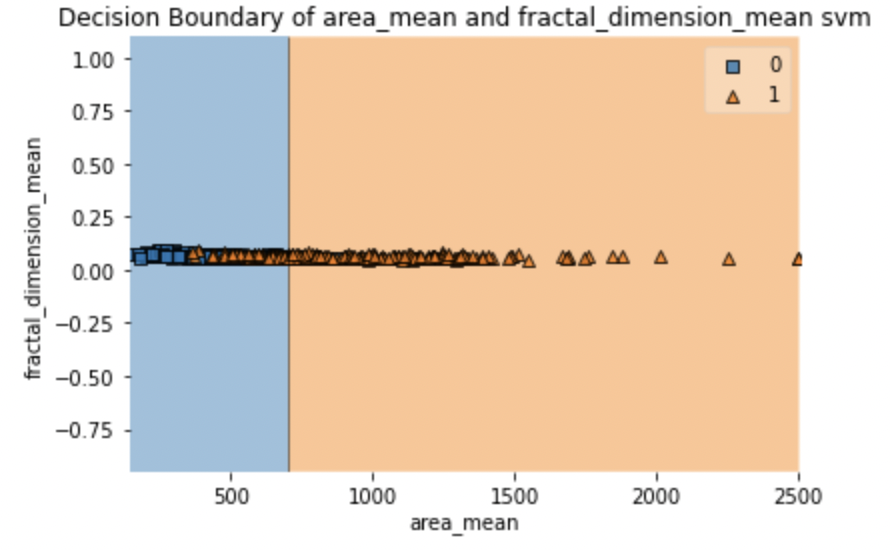
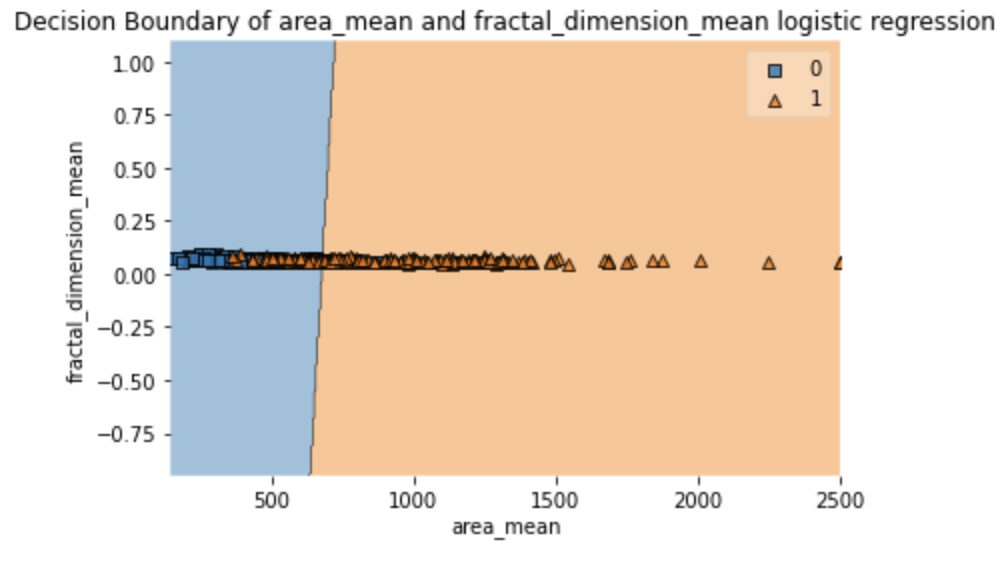
[](#D2L_fig_label_)

Figure 7: Decision Boundary of area\_mean and fractal\_dimension\_mean with Logistic Regression

Figure 8: Decision Boundary of area\_mean and fractal\_dimension\_mean with SVM

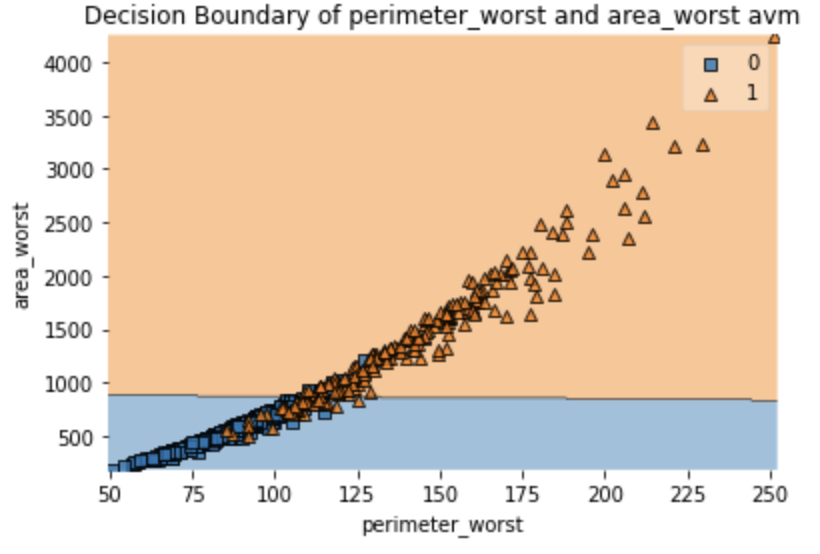
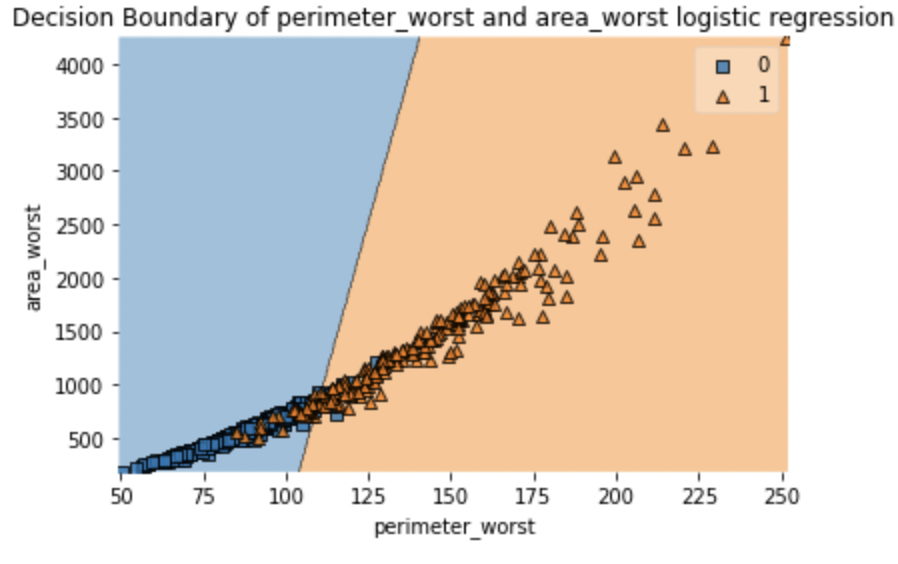
[](#D2L_fig_label_)

Figure 9: Decision Boundary of perimiter\_worst and area\_worst with Logistic Regression  
Figure 10: Decision Boundary of perimiter\_worst and area\_worst with SVM

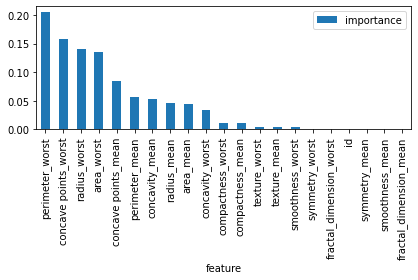
**Successful Modeling Attempts: 3 Separate Datasets & Random Forest Importance Features**

***Our 3 separate datasets mimic PCA properties and divide our data in the best way possible***

We were unable to use PCA to combine homogeneous variables, but that did not stop us from exploring our data further and experimenting ways in which we can do this. As our pre-processing steps explain, we divide our code into 3 separate data types (Mean & Worst, Mean, Worst). By doing so, we were able to examine our models in three different ways, allowing us to analyze our data further and find similarities between the dataset models that point out important conclusions. Although PCA was unhelpful, we were essentially able to mimic its properties and divide our data in the best way.

***Random Forest points out perimeter, concave points, and concavity as most important features***

Amidst our failed PCA and logistic regression attempt, we looked at other models like random forests to find important variables. Feature importance is based on a mean decrease in impurity (averaging across all the trees generated in the random forest). In order to calculate the average importance of each variable in our dataset, we first fit our X variables (all variables except the class variable) and y variable (diagnosis variable) training data into the Random Forest Classifier. Then we extracted the importance values for each column of the X variables and created a data frame to store those values and their column labels. Finally, we plot the importance of each variable in a bar plot as shown.

[](#D2L_fig_label_Importance Values for Mean & Worst Variables)  
Figure 11: Importance Values for Mean & Worst Variables

Notice that the most important features for the worst and mean tumors are perimeter, concave points, and concavity. These are also the explanatory variables used in our reduced decision trees, meaning the size and shape of the tumors are important in our dataset. The separate mean and worst importance values bar plot point to the same important explanatory variables (See Appendix 2, 3).

**Successful Models: Decision Trees, k-NN, Neural Networks**

**Decision Trees**

***Decision trees are the only successful models that have interpretability features***

Our code for decision trees can be explained in 3 parts: the building of the full tree, the reduced tree, and the accuracy score/confusion matrix. To build the full tree, we used the Decision Tree Classifier and fit the X and y training data. Then we used the plot\_tree() function to visualize the full tree (Appendix 4, 5, 6).

Building the reduced tree required more parameters. We used the Decision Tree Classifier but added constraints to avoid overfitting. Here is the list of popular constraint parameters we used:

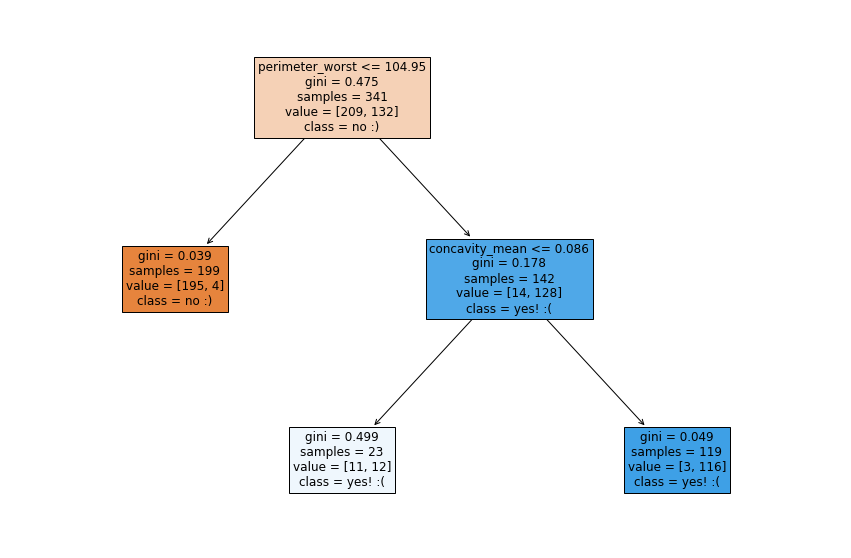
* max\_depth was set to 10 maximum number of splits
* min\_samples\_split was set to 40 minimum number of samples required to split a node
* min\_samples\_leaf was set to 20 minimum number of samples inside every leaf
* min\_impurity\_decrease was set to 0.01 minimum absolute ‘error’ improvement to justify a split
* criterion was set to ‘gini’, the default

Then, the X train and y train data were fit to the classifier. After defining the class names (‘no :)’ for benign/0 and ‘yes! :(’ for malignant/1), we visualized the reduced tree using the plot\_tree() function again.

In order to obtain the accuracy scores, confusion matrix, and true positive, false negative rates, we had to predict the y output for training and test data. Then we compared the y train data to the y predicted train data with the accuracy\_score() function. The y test and y predicted test data were compared in the same way to obtain the accuracy score. The confusion matrix was obtained using the same comparison parameters for the train and test data. We obtained the true positive rate with the recall\_score()function and the false negative rate was calculated using the confusion matrix.

When choosing a tree to deeply analyze, we chose the reduced trees over the full trees to keep the results more comprehensible for the reader and avoid overfitting. In addition, the explanatory variables used in the reduced trees were also the most important variables classified from the random forest classifier. Using the three different types of columns, we created three respective separate sets of full and reduced trees (mean & worst, mean, worst). The reduced trees and their explanations are provided below:

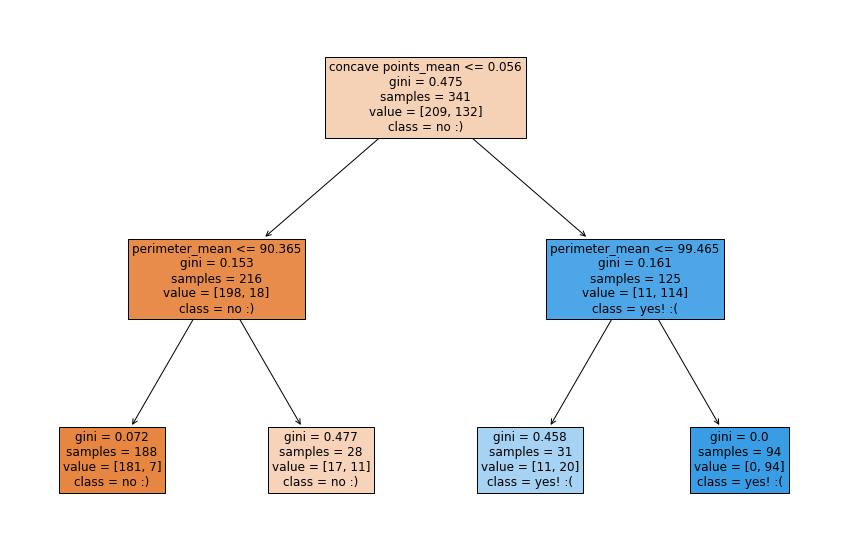
*Mean & Worst Description*

  
Figure 12: Reduced Tree on Mean & Worst Columns

The first node shows the splitting variable as the worst cancer cell’s perimeter being less than 104.95. The left branch indicates if the cell’s perimeter is less than or equal to 104.95, the tumor will be benign while if it is greater than 104.95, the tumor will be malignant. The second split’s variable is based on the cancer tumor having a concavity mean less than 0.086 and only applies to the right child node. However, regardless of whether the variable is true or false, the majority of each leaf ends with the cell being classified as malignant. Therefore, it can be concluded through this tree that once the perimeter surpasses a certain threshold, the cancer cell will most likely be malignant. The performance of this tree is as follows:

| Accuracy | True Positive Rate/Sensitivity | False Negative Rate |
| --- | --- | --- |
| 86.4% | 86.25% | 13.75% |

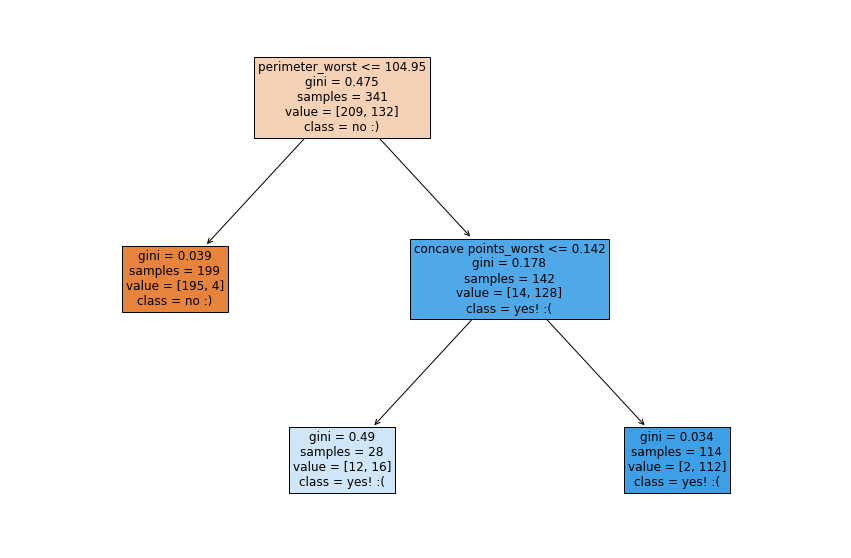
*Mean Tree Description*

  
Figure 13: Reduced Tree on Mean Columns

The first node shows the splitting variable as the cancer cell having average concave points less than 0.056. The left branch indicates if the cell has less than or equal to 0.056 concave points, the tumor will be benign while if it is greater than 0.056, the tumor will be malignant. The second split’s variable is regarding the perimeter mean of the cancer cell. Additionally, both branches under the left second child node both point to a benign tumor while both branches of the right child node point to malignant. This indicates that regardless of the perimeter mean if the mean concave point is greater than 0.056, the cancer cell is likely to be malignant. The performance of this tree is as follows:

| Accuracy | True Positive Rate/Sensitivity | False Negative Rate |
| --- | --- | --- |
| 91.2% | 85% | 15% |

*Worst Tree Description*

[](#D2L_fig_label_)  
Figure 14: Reduced Tree on Worst Columns

The first node shows the splitting variable as the worst cancer cell’s perimeter being less than 104.95 (similar to the mean and worst reduced tree shown before). The left branch indicates if the cell’s perimeter is less than or equal to 104.95, the tumor will be benign while if it is greater than 104.95, the tumor will be malignant. The second split’s variable is based on the worst cancer cell having less than or equal to 0.142 concave points and only applies to the right child node. However, regardless of whether the variable is true or false, the majority of each leaf ends with the cell being classified as malignant. Therefore, it can be concluded through this tree that once the perimeter surpasses a certain threshold, the cancer cell will most likely be malignant. The performance of this tree is as follows:

| Accuracy | True Positive Rate/Sensitivity | False Negative Rate |
| --- | --- | --- |
| 86.4% | 86.25% | 13.75% |

We noticed that throughout all the datatypes (mean and worst, mean, worst), the parameters that were chosen under the reduced tree constraints use the same variables: the tumor’s perimeter, concavity, and concave points. These are also the most important variables the random forest classified earlier.

The perimeter of the records shown in the worst tumors tells us that smaller perimeters under a certain threshold (104.95) are benign and above are malignant (Figure 14). This means a smaller tumor has a greater probability of being benign.

In addition, the worst column reduced tree (Figure 14) shows a tumor with a perimeter greater than 104.95, and concave points greater than 0.142 are highly likely to classify as malignant. This supports our theory that more concave points lead to a higher probabilistic chance of a tumor being malignant. However, concave points less than or equal to 0.142 were also classified as malignant, but with less certainty (12/28 incorrect classifications of benign compared to 2/114 on the other side).

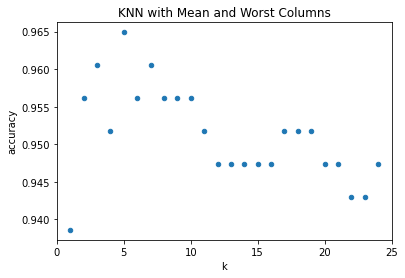
Lastly, the mean and worst column reduced tree (Figure 12) shows the worst tumors with a perimeter greater than 104.95 and mean concavity greater than 0.086 are highly likely to classify as malignant. This means greater concavity (more protrusion into the cell) leads to higher chances for a tumor to be malignant. Those records with concavity less than or equal to 0.086 are classified as malignant but should be inconclusive since the samples between yes and no are almost equal (11 no’s, 12 yes’s). This also explains the lower accuracy score.

**k-Nearest Neighbors**

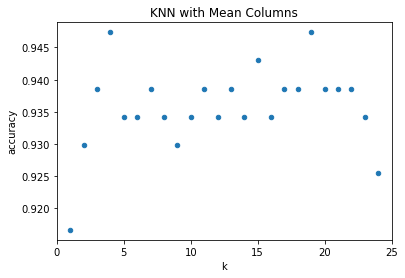
***Uniform weights k-NN algorithm yielded promising prediction results in testing set***

After attempting to understand the importance of each feature in our dataset, we decided to shift our focus to generating the best model for predicting malignancy. Because our dataset is not large and low on dimensionality, we implemented a k-Nearest neighbors algorithm where each data point is classified as whatever the predominant class is among the k-amount of nearby data points. Because each data point represents a distinct patient’s tumor with various features, we used the ‘uniform’ weight function as all points and features should be weighted equally.

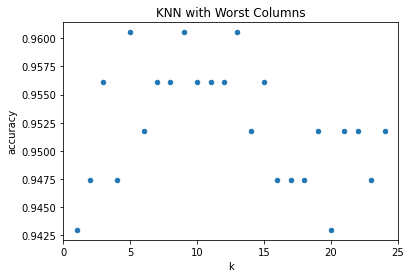
Similar to what we did in previous models, we ran three separate k-NN models with the “mean” columns only, with the “worst” columns only, and with the “mean” and “worst” columns combined. We standardized the training and validation features using the StandardScaler() function. Then we used a for loop of 25 iterations to test for the best k-value. Through examination of the plots of accuracy versus k-values in the three models (Figures 15, 16, 17), we chose a k-value of 5 with the model consisting of both “mean” and “worst” columns because it yielded the best accuracy score, True Positive Rate, and False Negative rate, which are all important indicators of an applicable malignancy prediction model.

[](#D2L_fig_label_)  
Figure 15: k-NN with Mean & Worst Columns

| Accuracy | True Positive Rate/Sensitivity | False Negative Rate |
| --- | --- | --- |
| 96.49% | 92.94% | 7.06% |

[](#D2L_fig_label_)  
Figure 16: k-NN with Mean Columns

| Accuracy | True Positive Rate/Sensitivity | False Negative Rate |
| --- | --- | --- |
| 94.74% | 88.24% | 11.76% |

[](#D2L_fig_label_)  
Figure 17: k-NN with Worst Columns

| Accuracy | True Positive Rate/Sensitivity | False Negative Rate |
| --- | --- | --- |
| 96.05% | 91.76% | 8.24% |

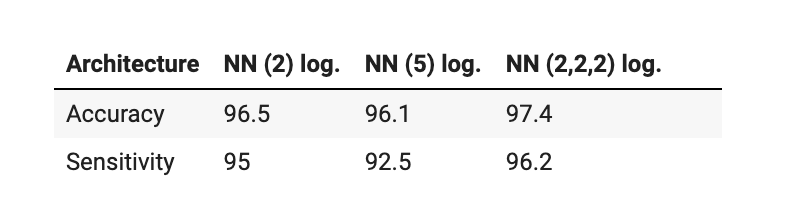
**Neural Networks**

***Neural Network achieved the best results despite lack of knowledge on feature relationships***

After running k-NN and decision trees to diagnose patients’ cancer conditions, we wanted to explore more models for this problem. Neural networks have a great predictive ability to capture complicated, non-linear relationships between variables. Similar to the prior models, we explored 3 different neural network models with 3 different sets of data: a hidden layer with 2 nodes, a hidden layer with 5 nodes, and 3 hidden layers with 2 nodes. As a result, we ran a total of 9 neural networks to find the optimal weight that yields the best prediction.

Before running the model, we had to preprocess the X variables and put them through different activation functions in order to generate a Y output. The first step was to preprocess the X variables. In this step, we turned the categorical variable ‘diagnosis’ to dummy values, and scaled all other numerical variables to a 0-1 scale, using a min-max normalization function. To follow, those inputs were run into the 4 different activation functions: Logistic, Linear, Tanh, and Relu under sklearn’s neural network package called the MLPRegressor. Under these conditions, we found that the logistic model works best with our predictions. However, the diagnosis resulted in numerical values with the logistic model. In order to combat this, we transferred the numerical values to binary values through the NumPy package (i.e. ‘y\_pred\_train = np.where(y\_pred\_train > 0.5, 1, 0)).

Neural networks turned out to be a great metric for predicting tumor diagnosis. It had the highest accuracy and sensitivity rate compared to other models (See Figure 17, the last model had an accuracy rate of 97.4% and a sensitivity rate of 96.2%). The high accuracy and sensitivity rate show that the neural network has the strongest ability to find all the positive samples and predict the correct results. However, there were still some drawbacks to this model. The relationship between predictors and outcomes remains unanswered, making it hard to identify key variables that determine breast cancer classification. In addition, neural network models require large datasets for prediction, while our data size was relatively small.

[](#D2L_fig_label_)  
Figure 17: The Accuracy and Sensitivity Rate of Neural Network with Worst Columns

**Conclusion**

***Despite lacking the ability to explain key variables, the neural networks model had the best performance relative to accuracy, TPR, and FNR***

Our main purpose for this project was a trial-and-error process to find the optimal model to classify cancer malignancy given the different features of the tumor. Upon reviewing our model results (See Table 1), we chose to focus on four important indicators which include Accuracy, True Positive Rate (TPR), False Negative Rate (FNR), and whether the model can identify key variables. The most undesirable outcome would be if we falsely predict a tumor to be benign when it should actually be malignant. Because wrong predictions in a cancer malignancy model can potentially cost patients’ lives, we have to ensure the chosen model achieves the highest TPR and the lowest FNR. Therefore, we selected Neural Networks as our best model because it had a 97.4% accuracy, a 96.2% TPR, and a 3.8% FNR. However, because both k-NN and neural networks lack the ability to explain key variables, we would have to utilize the Decision Trees model for knowledge of feature importance.

| Model | Accuracy | Sensitivity/TPR | FNR | Can Identify Key Variables |
| --- | --- | --- | --- | --- |
| Decision Trees | 86.4% | 86.3% | 13.7% | O |
| k-NN | 96.5% | 92.9% | 7.1% | X |
| Neural Network | 97.4% | 96.2% | 3.8% | X |

[Table 1:](#D2L_table_label_) Model Results

***To overcome limitations that were encountered, possible exit opportunities include further data exploration by collaborating with hospitals to expand available data and resources***

Regardless of the promising results, our models encountered several limitations and constraints during the course of data exploration. Firstly, our dataset being small and only containing 500 entries could lower our precision and increase the risk of errors. In addition, we were unable to test our models with new patient records. Due to the medical complexity of the variables, we were unable to insert new numbers without proper medical examinations of tumor features. Lastly, our better performing models –such as the neural network– only provide accurate predictions but do not explain the depth and detail of the specific variables and their meanings. In order to overcome these limitations, we can further improve our research by taking the following measures:

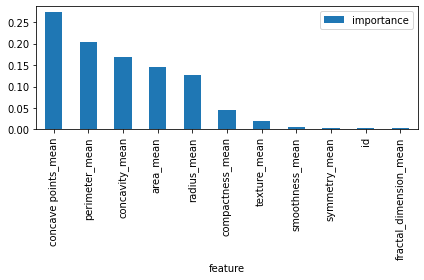
1. Collaborate with hospitals to develop our models by expanding the data records and predicting real patient data.

2. Further data exploration with other models that have explanatory features similar to the Decision Tree model.

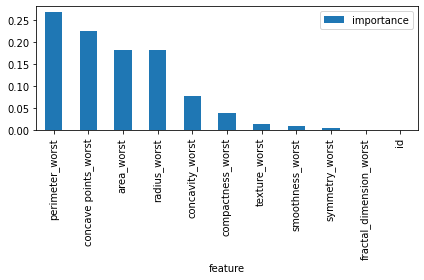
**Appendix**

| **Y Variables** | |
| --- | --- |
| Diagnosis | Classify patients’ health condition, M = malignant, B = benign. |
| **X Variables** | |
| Nucleus | The center of the cell that contains the cell's DNA. |
| Radius | Mean of distances from the center to points on the perimeter |
| Texture | Measured by finding the variance of the grayscale intensities in the component pixels. Generally, a more homogenous texture corresponds to reduced risk while more coarseness denoted increased risk. |
| Perimeter | Measured by the total distance between snake points. |
| Area | Measured by counting the number of pixels on the interior of the snake and adding one-half of the pixels in the perimeter. |
| Smoothness | Local variation in radius lengths, measuring the difference between the length of a radial line and the mean length of the lines surrounding it |
| Compactness | Perimeter and area combined to give a measure of the compactness of the cell nuclei using perimeter/area |
| Concavity | The measurement shows the severity of concave portions of the contour. Draw chords between non-adjacent snake points and measure the extent to which the actual boundary of the nucleus lies on the inside of each chord. |
| Concave points | Number of concave portions of the contour. The concave points measure how many indentations are on the surface of the nucleus. |
| Symmetry | Measured using the length difference between lines perpendicular to the major axis to the cell boundary in both directions |
| Fractal dimension | A [fractal](https://en.wikipedia.org/wiki/Fractal) dimension is an index for characterizing fractal (geometric) patterns or sets by quantifying their complexity as a ratio of the change in detail to the change in scale. |

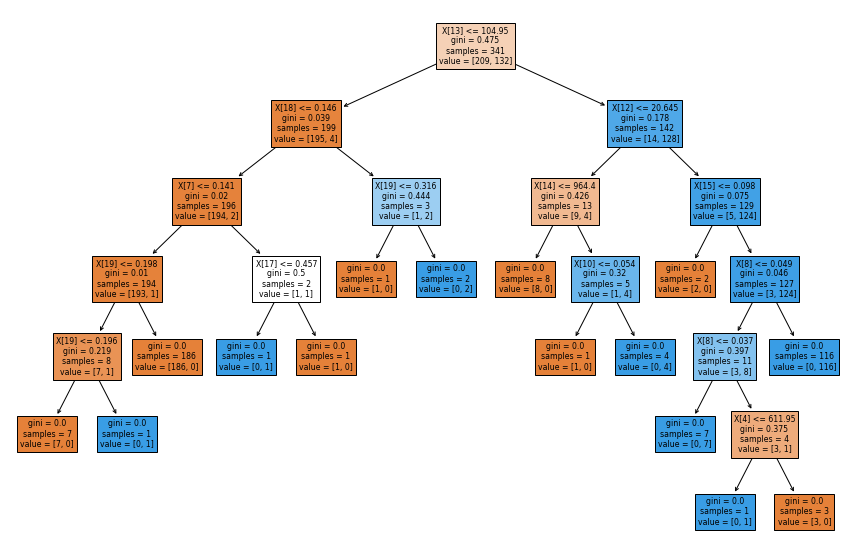
Appendix 1: Explanation of Variables

****

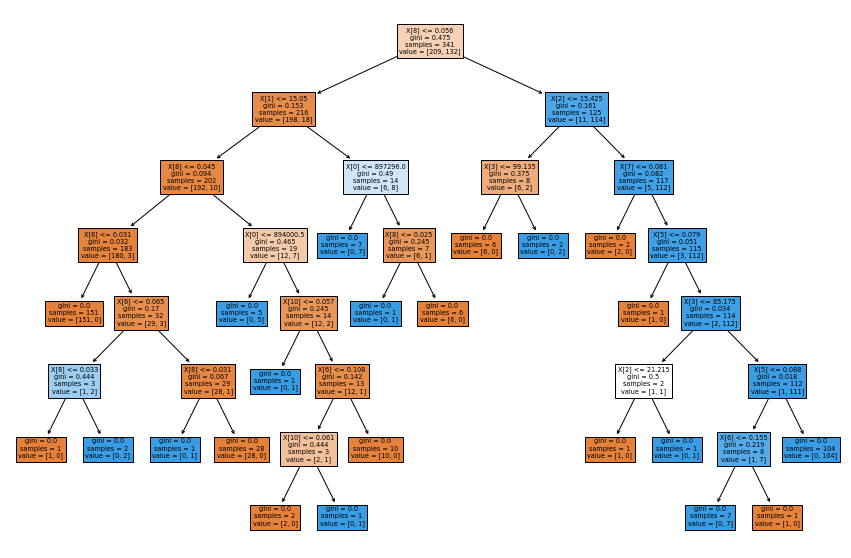
Appendix 2: Importance Values for Mean Variables



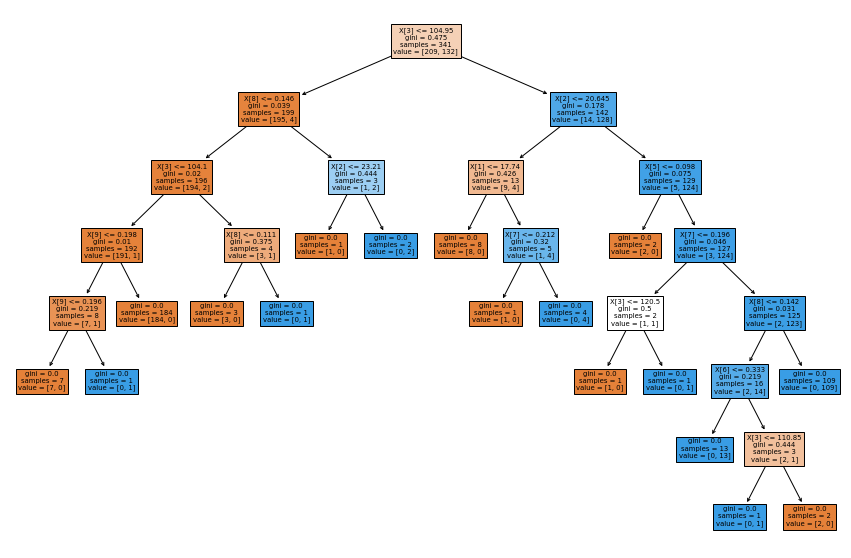
Appendix 3: Importance Values for Worst Variables



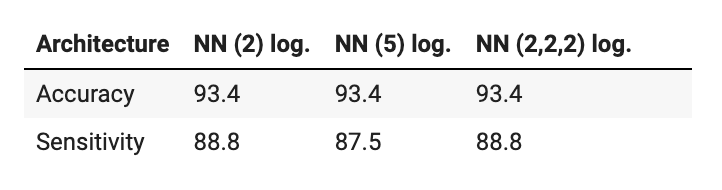
Appendix 4: Full Decision Tree with Mean and Worst Columns



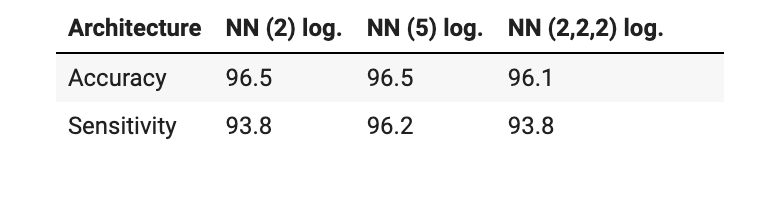
Appendix 5: Full Decision Tree with Mean Columns



Appendix 6: Full Decision Tree with Worst Columns

[](#D2L_fig_label_)

Appendix 7: The Accuracy and Sensitivity Rate of Neural Network with Mean Columns

[](#D2L_fig_label_)

Appendix 8: The Accuracy and Sensitivity Rate of Neural Network with Mean & Worst Columns

1. “Cancer.” World Health Organization. World Health Organization. Accessed May 1, 2022. <https://www.who.int/news-room/fact-sheets/detail/cancer#:~:text=Cancer%20is%20a%20leading%20cause,and%20rectum%20and%20prostate%20cancers>. [↑](#footnote-ref-0)
2. “Breast Cancer Statistics: How Common Is Breast Cancer?,” American Cancer Society, accessed May 1, 2022, <https://www.cancer.org/cancer/breast-cancer/about/how-common-is-breast-cancer.html>. [↑](#footnote-ref-1)