Breast Cancer Diagnosis Classification Analysis

Austin Lackey, Ethan Powers and Danny Laposata

December 9th, 2022

Contents

1 Introduction (Should we do an Abstract?)

1.1 Background

In our analysis, we will be using the Breast Cancer Wisconsin (Diagnostic) Data Set from Kaggle. This data set contains `569 observations` of breast cancer cells with `32 variables` describing each cell. Some of the variables include characteristics like `radius`, `texture`, `area`, `perimeter` of the cell.

The goal of our analysis is to predict whether a cell is benign or malignant based on the `32 variables`. A cell is considered benign if it is not cancerous and malignant if it is cancerous. Normally in most machine learning models, we do our best to train the model to reduce the overall error rate. While this is an important goal, our group was more concerned with the `Type-II error rate`. By reducing the `Type-II error rate`, we can ensure that we are not making the mistake of classifying a malignant cell as benign. This is important in the world of Oncology because if a malignant cell is classified as benign, it could lead to a patient not receiving the proper treatment. Whereas if a benign cell is classified as malignant, the patient may be alarmed, but a false alarm is better than a missed diagnosis.

In order to achieve our goal, we conducted the following steps:

1. Data Cleaning
2. Explatory Data Analysis
3. Classification Analysis
4. Regression Analysis
5. Overall Analysis Summary

1.2 Data

In order to properly train and test our models, we first had to split the data into a training and test set. We decided to use a `60/40 split` for our training and test data. This allows us to allocate more data to the training set, which will allow us to train our models more effectively. We also decided to remove the `ID` column from the data set because it was not relevant to our analysis.

1.3 Variable Descriptions

After removing the `ID` column, we were left with `31 variables`. For this project our response variable is `diagnosis` as this variable tells whether an observation is Malignant (cancerous cells): `M` or Benign (noncancerous cells): `B`. The predictor variables make up the remaining 30 columns; however there are only actually 10 variables that each have 3 measurements. These measurements are `mean`, `standard error (SE)`, and `worst` (the mean of the 3 largest values of a variables).

Below is a brief explanation of what each of the 10 variables represent:

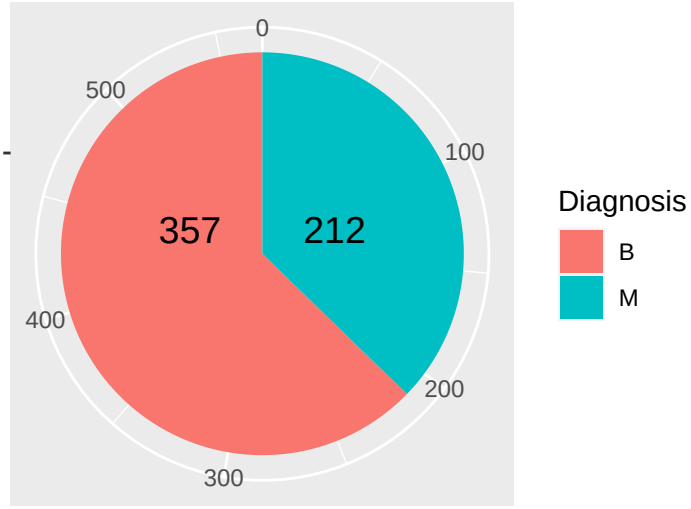
- Radius: Average Distance from cell center to cell perimeter
- Texture: Standard deviation of gray-scale values; brightness of pixel of cell
- Perimeter: Distance around nucleus boundary
- Area: Area of the nucleus
- Smoothness: Variation in cell's radial lengths
- Compactness: The Perimeter^2/Area
- Concavity: Size of the indention in nucleus boundary
- Concave Points: Number of points on indented section of nucleus boundary
- Symmetry: Deviation of the nuclei shape from the ideal measurement
- Fractal Dimension: Measurement of irregularity in nucleus boundary

2 Explatory Data Analysis

To begin our analysis, we first wanted to get a better understanding of the data so we could properly prepare it for our models. As you can see in Figure A below, we have more information regarding the benign cells

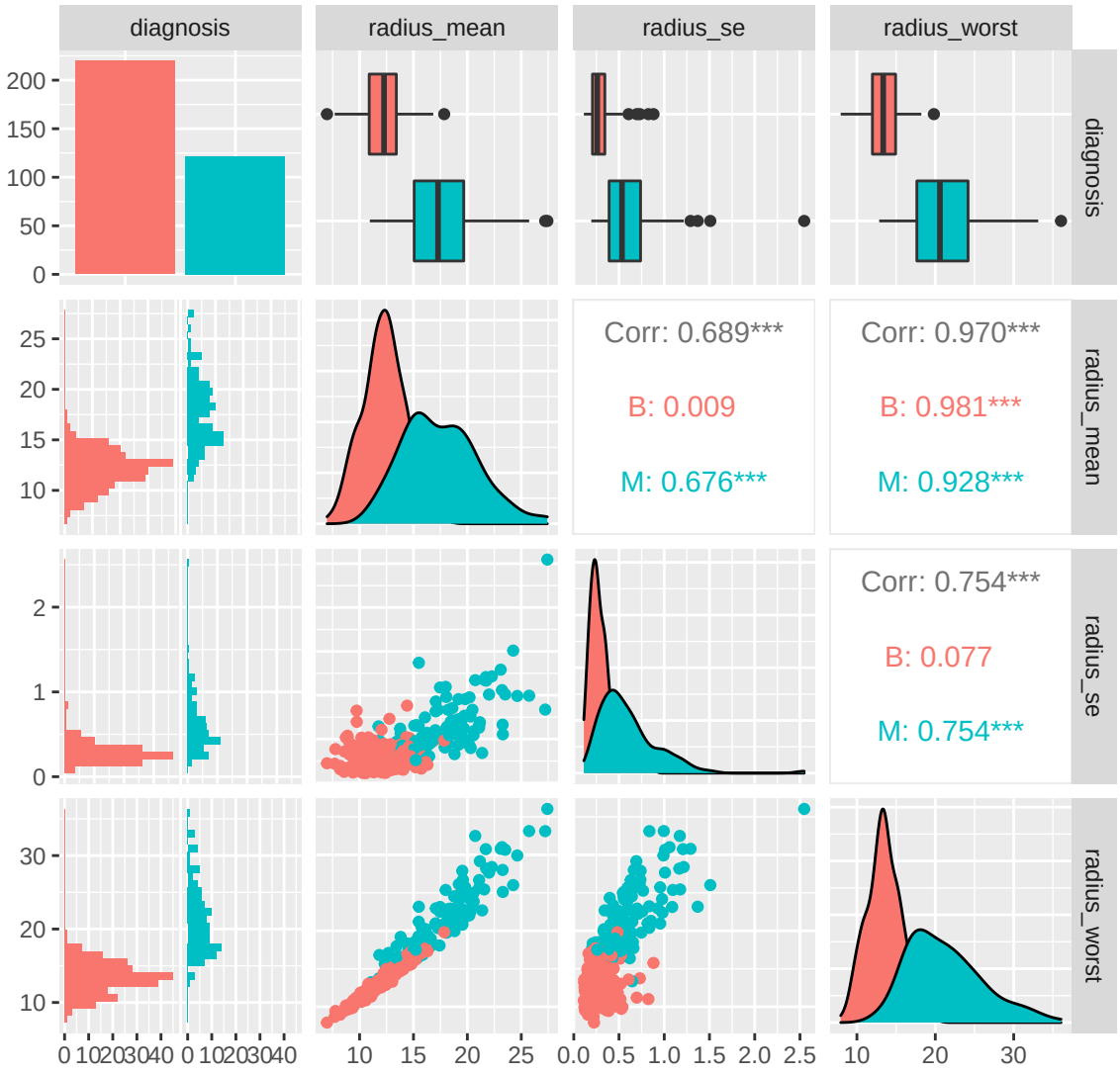
than the malignant cells. To be more specific, we have 357 benign cells and 212 malignant cells. Since we are worried about the `Type-II error rate`, we would want more information regarding the malignant cells in an ideal world. However since this is not the case, is important to note because it could lead to a bias in our model if we do not take this into account.

Figure A: Malignant vs Benign Cell Counts



Another important thing to note about the data is the variables included. There are three different measurements for the ten predictor variables: mean value, standard error, and "worst" value (mean of 3 largest measurements). As shown the correlation plot below, these three measurements are all correlated to each other, but still hold a strong relationship with the diagnosis response variable. The relation between the mean value and "worst" value is almost completely correlated which makes sense as both are averages however, the "worst" value only uses the three largest measurements. This is important to know when trying to interpret feature effect, importance and selection.

Figure B: Pairwise Scatterplots of Features



3 Classification Analysis

During our classification analysis, we attacked `Type-II error` in two different ways. The first thing we did was use different paramters for each model and to figure out which parameters yield the lowest `Type-II error rate`. However, testing many different parameters can be time consuming and computationally expensive. We also did a different approach by tuning his models to be more sensitive to Malignant cells. This allows us to reduce the `Type-II error rate` by classifying more cells as Malignant. However, on the downside, this also increases the `Type-I error rate` by classifying more cells as Malignant. By harnessing both approaches, we were able to reduce the `Type-II error rate` to its fullest while also maintaining a good `Type-I error rate`.

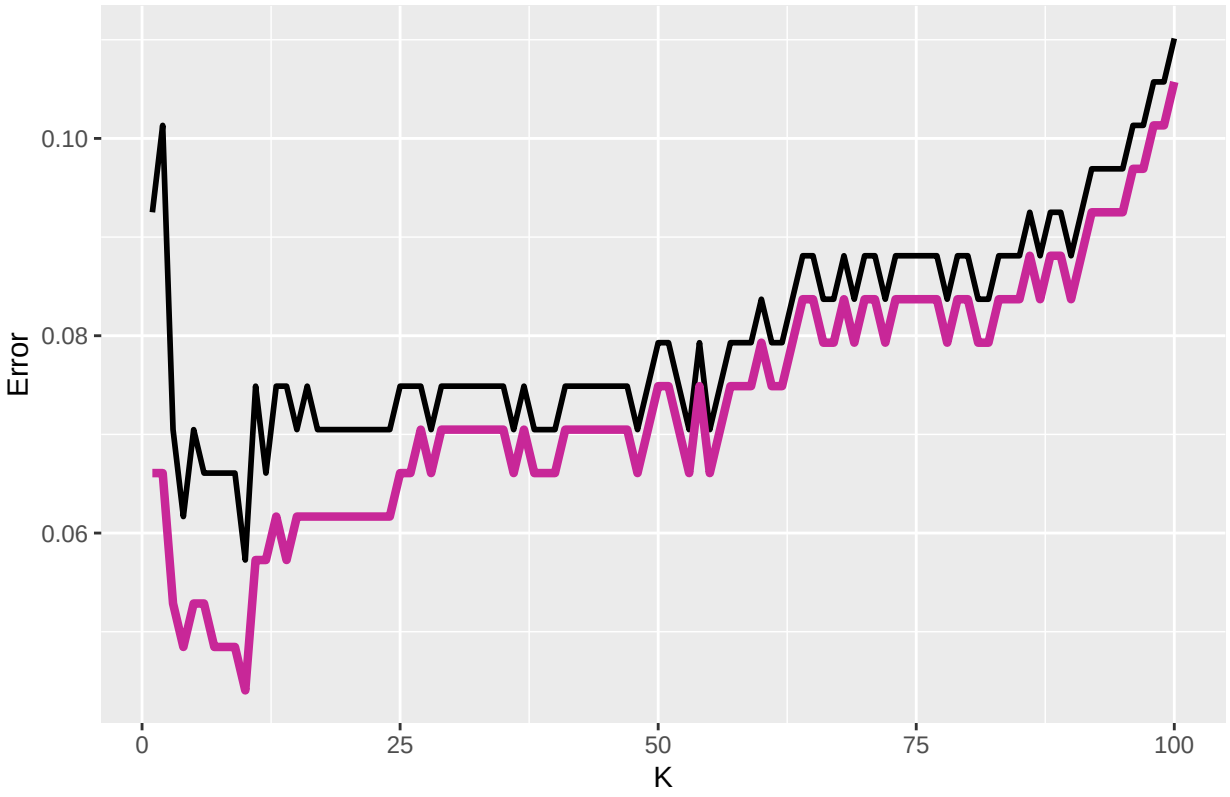
3.1 Austin's Classification Proccess and Analysis

The following models used were tuned using a `10-fold cross validation` method that was repeated 10 times. Cross validation is method that is used to train a model on a subset of the data and then test the model on the remaining data. This process is repeated `10` times with each subset of data being used as the test set once. By using this method we are able to effectively train our model while also testing it on data that it has not seen before.

3.1.1 KNN Model

The first model that we used was a `K Nearest Neighbor's model`. In order to tune this model, we used the `tuneGrid` parameter to test different values of `k`. We tested `100` values of `k` ranging from `1` to `100` and then plotted the `Type-II error rate` (in purple) as well as the overall error rate (in black) for each value of `k` as shown in Figure C below. The model with the lowest `Type-II error rate` was the model with `k = 10`. Any value of `k` greater than `10` resulted in a higher `Type-II error rate` as well as a higher overall error rate. This can be attributed to the fact that the model is overfitting the data.

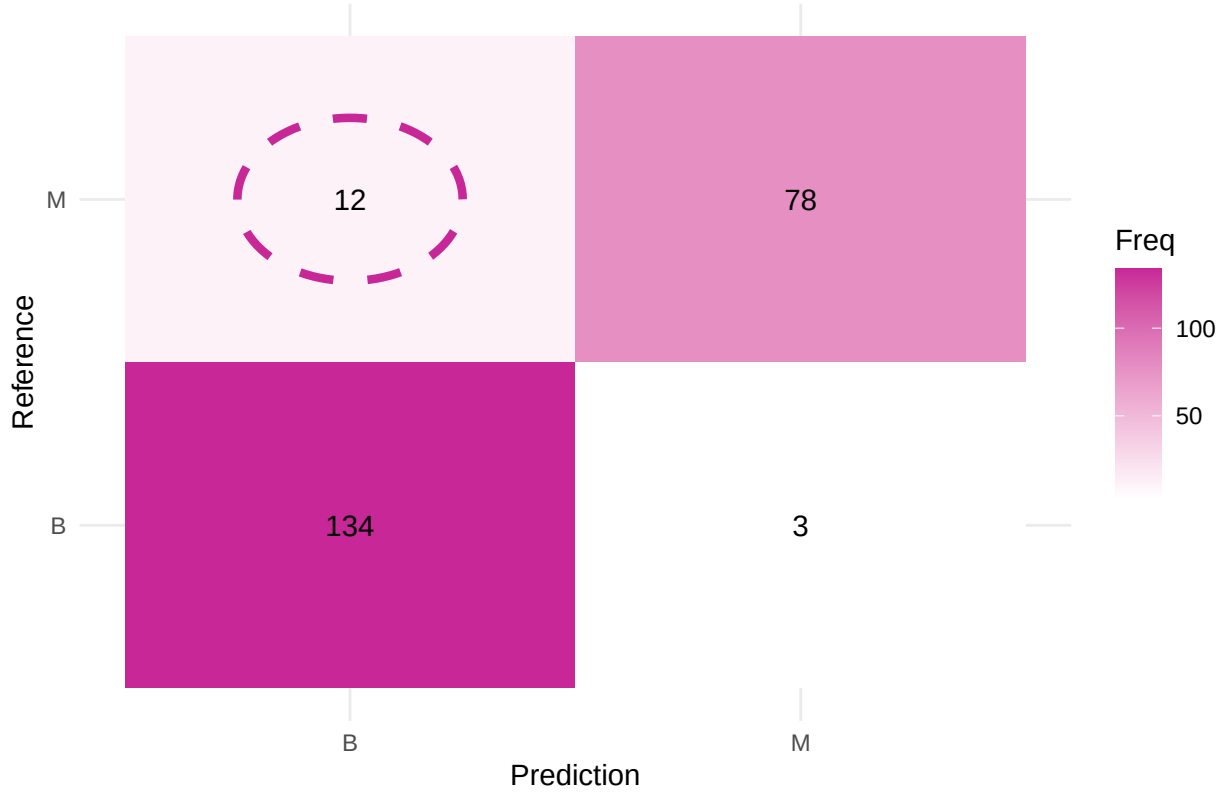
Figure C: KNN Type II and Overall Error



3.1.2 Tuned KNN Model

Using the information from the previous plot, we were able to tune our model to have a `k` value of `10`. This tuned model resulted in a `Type-II error rate` of 0.0440529 and an overall accuracy of 0.9339207. The confusion matrix for this model is shown in Figure D below. As you can see out of the total 342 training samples, 12 were misclassified as benign when they were actually malignant.

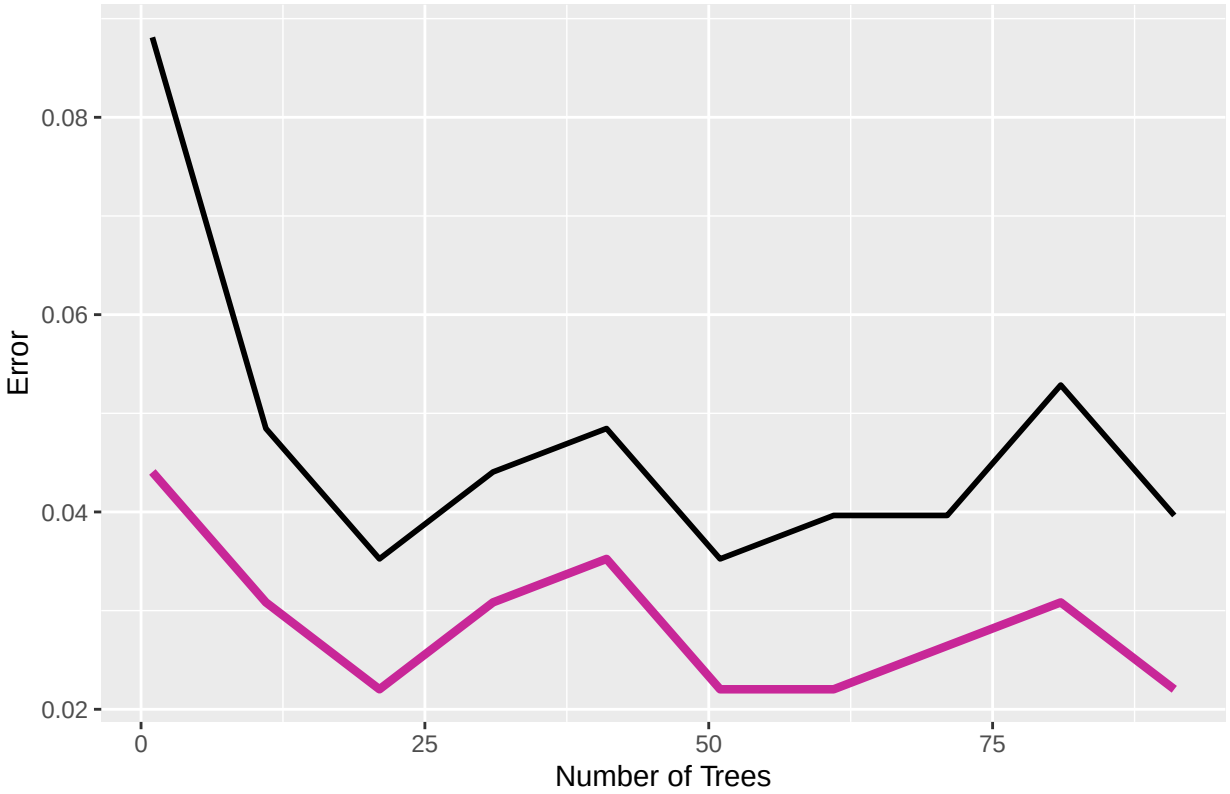
Figure D: KNN Confusion Matrix



3.1.3 Random Forrest Model

The second model that we used was a `Random Forrest model`. In order to tune this model, we used the `tuneGrid` parameter to test different numbers of `trees`. We tested `10` values of `trees` ranging from `1` to `100` and then plotted the `Type-II error rate` (in purple) as well as the overall error rate (in black) for each value of `trees` as shown in Figure E below. The model with the lowest `Type-II error rate` was the model with `trees` = 21. Any value of `trees` greater than 21 resulted in a higher `Type-II error rate` or a higher computation time for the same `Type-II error rate`.

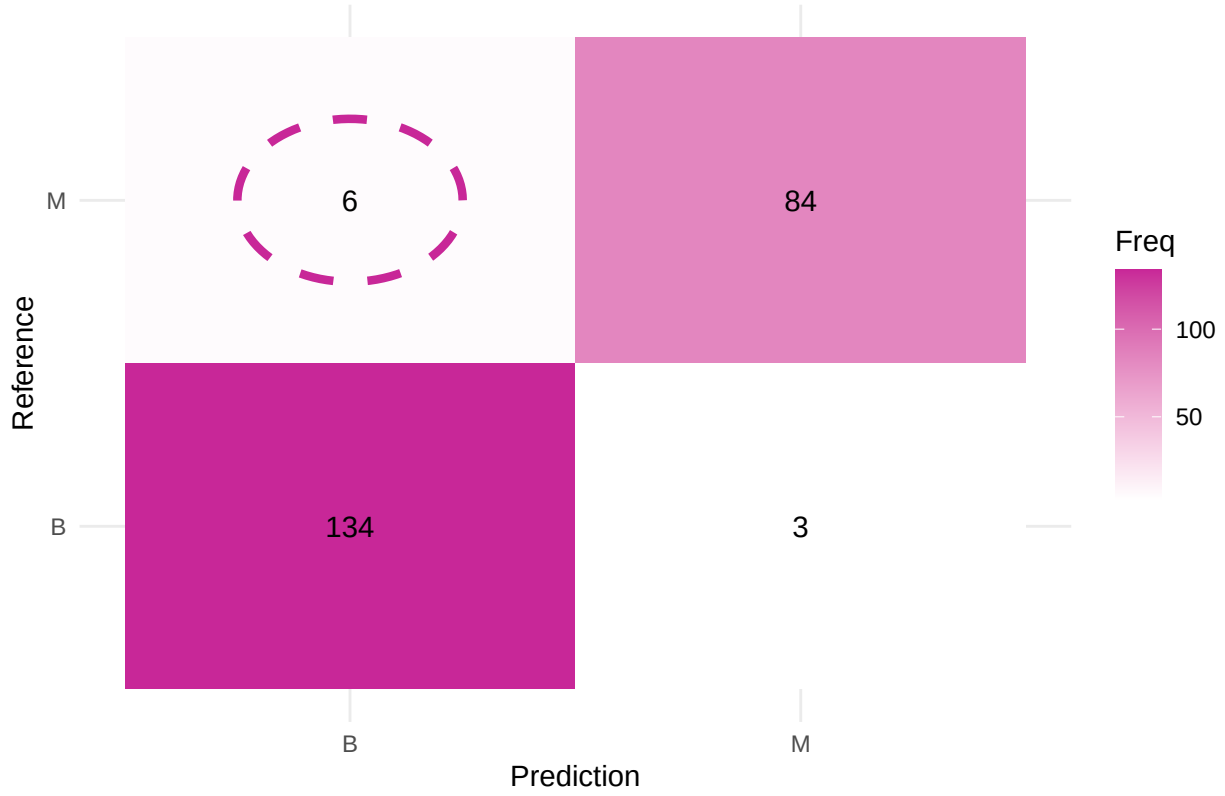
Figure E: Random Forrest Type II and Overall Error



3.1.4 Tuned Random Forrest

Using the information from the previous plot, we were able to tune our model to have a `trees` value of 21. This tuned model resulted in a `Type-II error rate` of 0.0220264 and an overall accuracy of 0.9603524. The confusion matrix for this model is shown in Figure F below. As you can see out of the total 342 training samples, 5 were misclassified as benign when they were actually malignant. And we were able to further reduce our `Type-II error rate` from 0.0440529 to 0.0220264 when compared to the KNN model.

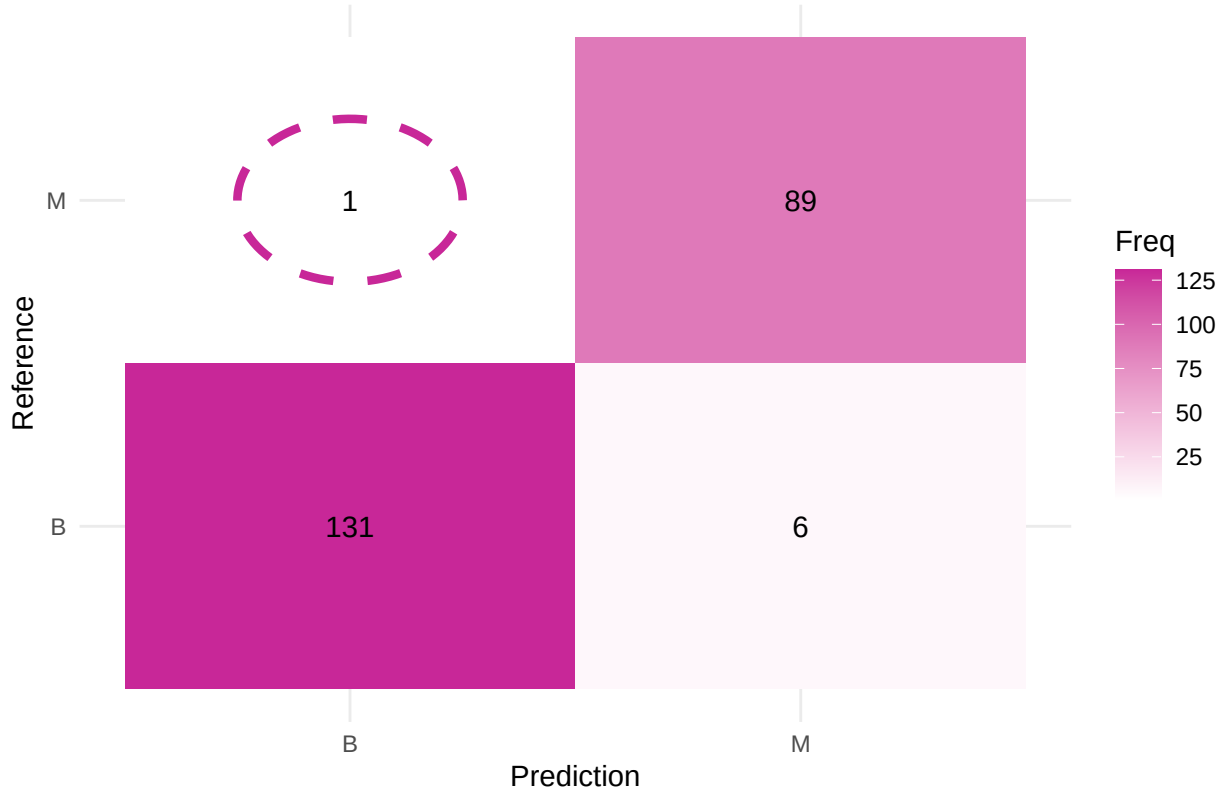
Figure F: Random Forrest Confusion Matrix



3.1.5 Radial Support Vector Machine

The third model that we used was a `Radial Support Vector Machine`. To begin we used a basic model with the default parameters and then tuned the model using the `tuneGrid` parameter. The basic SVM model resulted in a `Type-II error rate` of 0.0044053 and an overall accuracy of 0.969163. The confusion matrix for this model is shown in Figure G below. As you can see out of the total 342 training samples, 1 was misclassified as benign when they it was actually malignant. This is a great improvement in our `Type-II error rate` when compared to the other models.

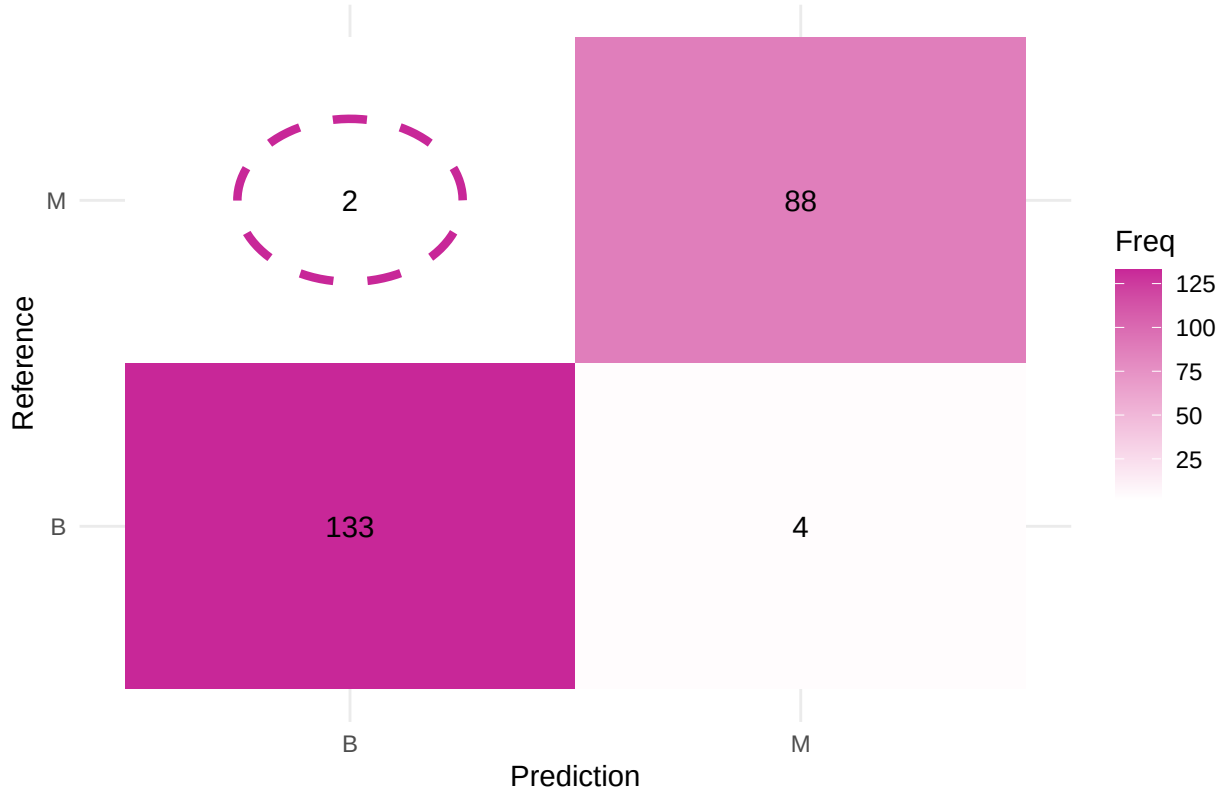
Figure G: Radial SVM Confusion Matrix



3.1.6 Tuned Radial Support Vector Machine

The final model that we used was a tuned `Radial Support Vector Machine`. The tuned model resulted in a `Type-II error rate` of 0.0088106 and an overall accuracy of 0.9735683. The confusion matrix for this model is shown in Figure H below. Out of the total 342 training samples, 2 were misclassified as benign when they were actually malignant. As you can see, our `Type-II error rate` was actually increased to 0.0088106 when compared to the basic SVM model. Since this svm model was tuned using carets `tune.svm` function, the overall error was reduced, but this resulted in a slightly higher `Type-II error rate`. For this reason, our basic model is actually better than the tuned model when it comes to meeting our goal of reducing the `Type-II error rate`.

Figure H: Tuned Radial SVM Confusion Matrix

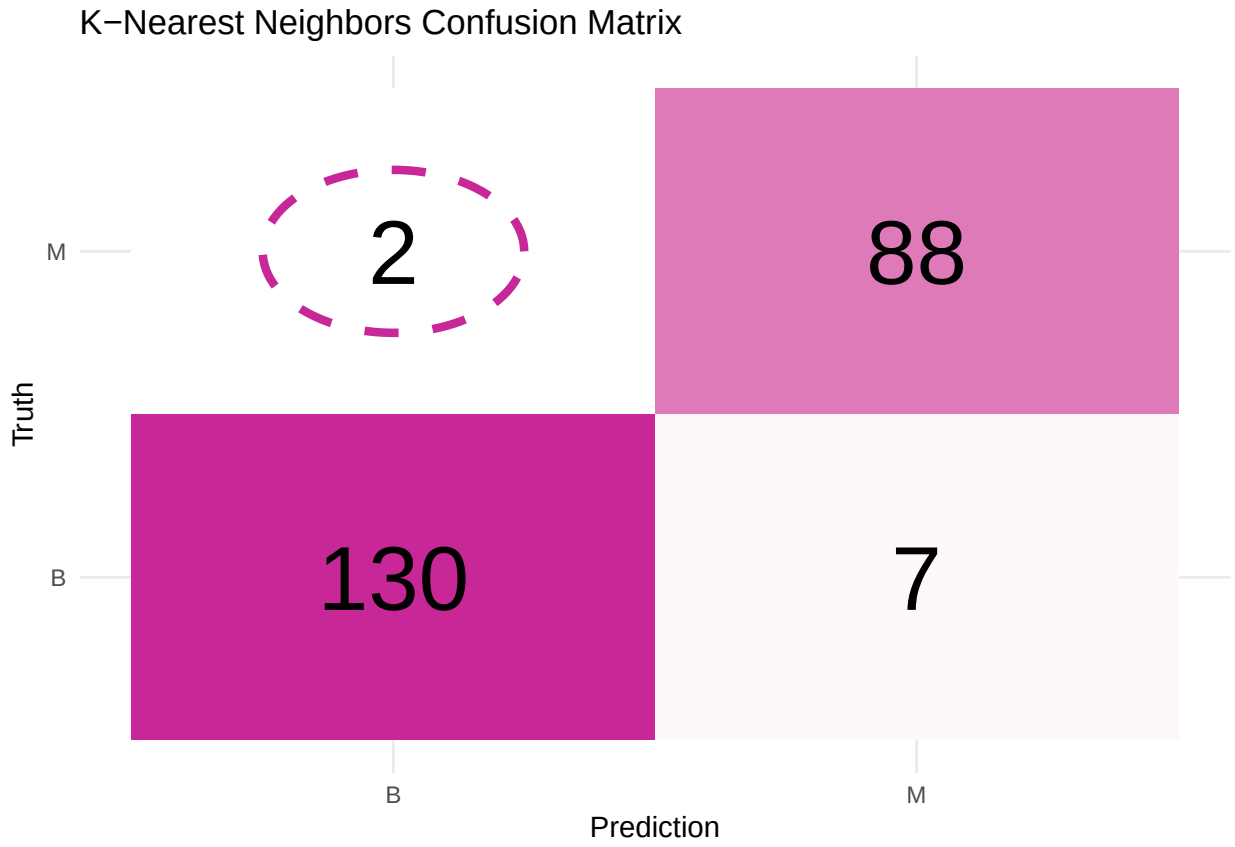


3.2 Dannys's Classification Proccess and Analysis

As the main purpose of this analysis is to reduce Type II error, we wanted to try different ways of recuding this within similar models. The following models all use classification to classify a tumor as benign or malignant however they have a lower prediction cutoff of 25% malignant to bias our results towards more malignant. For example, if a tumor is only 30% likely to be malignant, we still would classify that tumor as malignant to avoid false negatives. While this does lead to higher Type 1 error and sometimes lower overall accuracy, we aren't as worried about the false-positives. We fit a number of models on the training data using this approach, including Logistic Regression, K-Nearest Neighbors, Quadratic Discriminant Analysis (QDA), Random Forest, xgBoost, and lastly a support vector classifier. While all the most models performed quite decently, there were a few that stood out.

3.2.1 K-Nearest Neighbors

K-Nearest Neighbors is a classifying method that estimates the bayes classifier using the closest training points to the test point that is being classified. It uses a neighbor parameter that is a big factor in determining the classifier. For this model, we used 10-fold cross-validation to tune the neighbor parameter based on the highest accuracy. Once the neighbors parameter was tuned to 7, the model was fit and prediction ensued. The confusion matrix showcasing the classification prediction of this model is below.

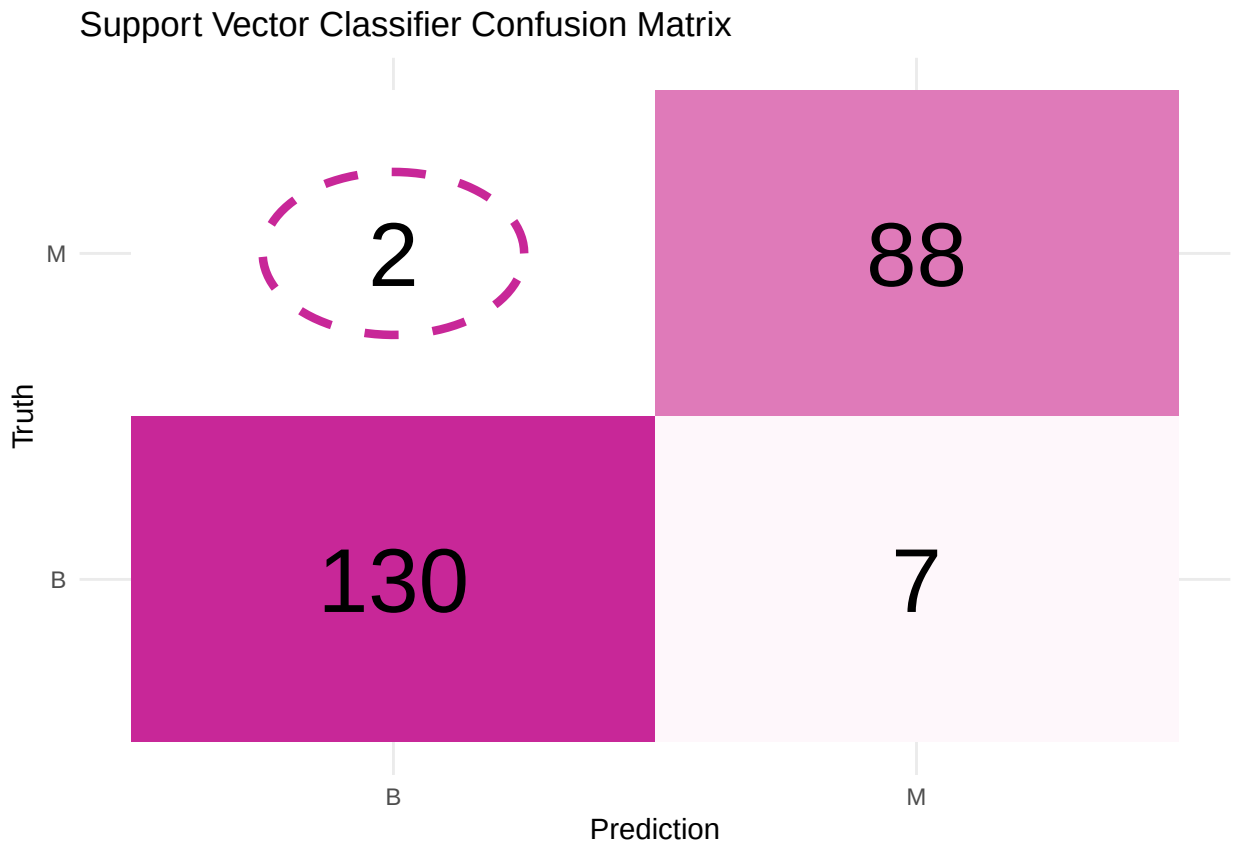


As shown in the confusion matrix, this KNN model only predicted two false-negatives out of 223 test observations and 90 true test malignant tumors. The KNN model predicted with an overall accuracy of 96.04. While this model did perform quite well, we still want to improve these results.

3.2.2 Support Vector Classifier

A support vector classifier uses a softened margin classifier to classify the observations but with less sensitivity than a maximal margin classifier. The support vector classifier will not necessarily classify all training points

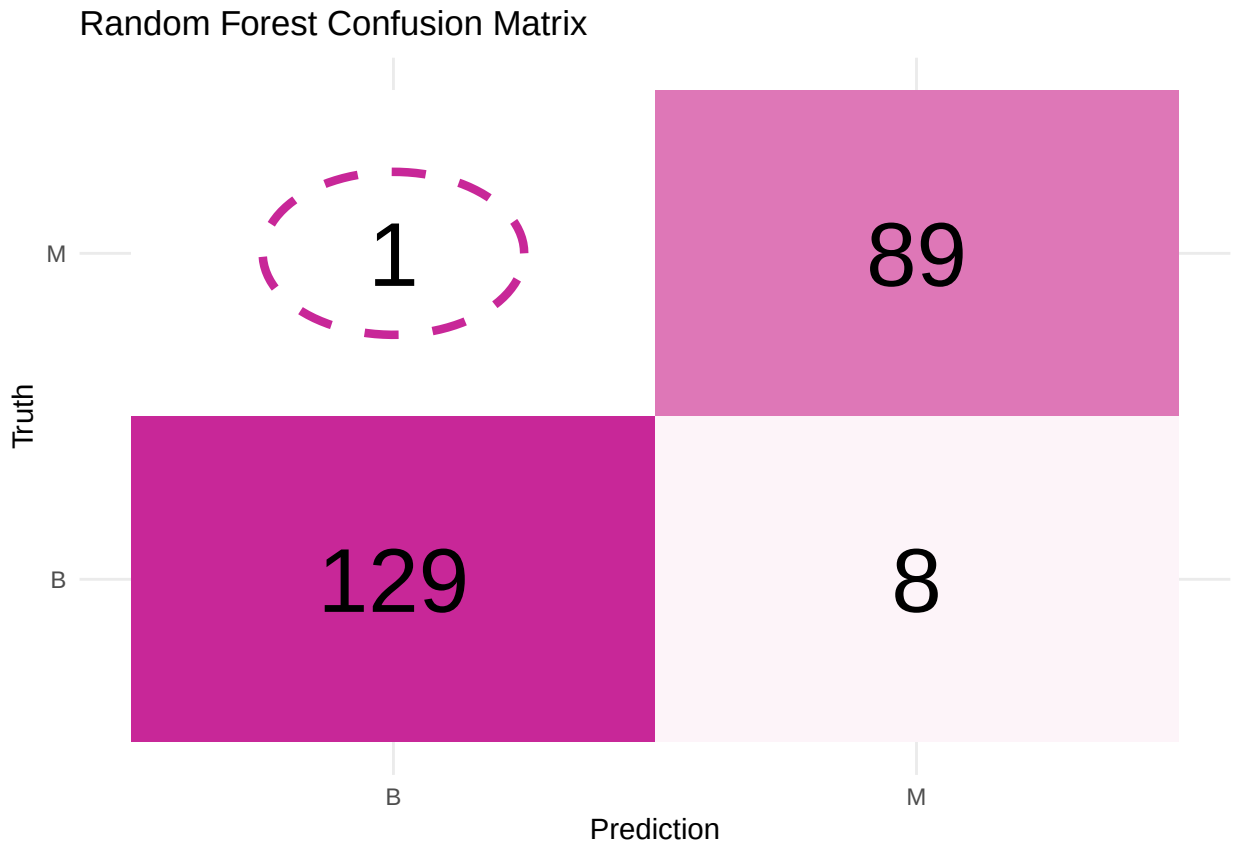
accurately, but this allows for better test prediction. We tuned our support vector classifier to take a cost value of 1 using 10-fold cross-validation. The confusion matrix showcasing the prediciton accuracy of the support vector classifier model can be seen below.



As shown above, the support vector classifier predicted the exact same as the KNN model. With only 2 false negatives and an overall accuracy of 96.04, this model was not an improvement from the KNN model, but it is reassuring to see consistent high accuracy amongst different models. The last model we will look at is a Random Forest model.

3.2.3 Random Forest

Random Forest creates many decision trees from bootstrapped sample training points and uses a subset of predictors for each tree. The main difference between Random Forest and bagging decision trees is this feature subset that helps decorrelate the trees. For this Random Forest model, a default of 500 trees was used as tuning did not improve the model. The confusion matrix showcasing the prediction accuracy of the Random Forest model is shown below.



As shown above, the Random Forest model predicted the best out of all our models according to the standards we set. With only one false-negative, this is the lowest out of all the models and with an overall accuracy of 96.04 it holds up against the other models and does not gain too much Type I error. This confusion matrix portrays the predicition cutoff very well. Because the model biases towards classifying as malignant, there is much higher Type I error than Type II error, which is exactly what we were aiming for.

4 Ethans's Regression Proccess and Analysis

While Austin and Danny did Classification Analysis on `Type-II error rate`, I decided to do a Regression Analysis using the variable importance from their analysis's to compute the `training` and `test` error rate of those variables. This is to see how much error exists in a variable that is important in calculating if a cell is Malignant or Benign. However, I will be focused on `test` error in my results as its more important to see how our results will be on future data rather than what we have already trained. The regression models that I will be using are:

- `Linear Regression`
- `Ridge Regression`
- `Lasso Regression`

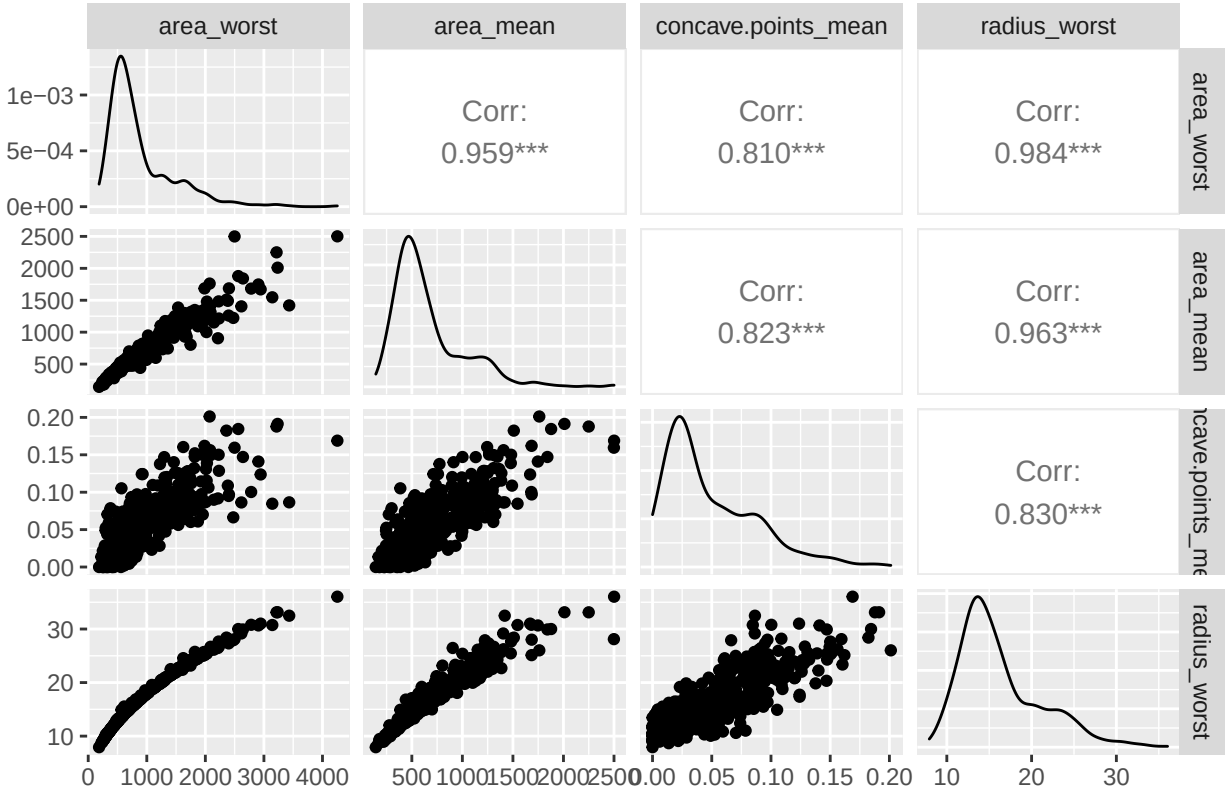
4.1 Variables

In order to find the variables to run the Regression Analysis on, I had Austin and Danny run a variable importance plot on the data they collected and took the top 5 variables from each of their results, which are listed below.

- Austin Top 5 Importance Variables:
    1. *`area_mean`
    2. *`area_worst`
    3. *`concave.points_mean`
    4. `texture_mean`
    5. *`radius_worst`
- Danny Top 5 Importance Variables
    1. `concave.points_worst`
    2. *`concave.points_mean`
    3. *`area_worst`
    4. *`area_mean`
    5. *`radius_worst`

After I got both Austin's and Danny's top 5 importance variables, I noticed that there were 4 common variables in each of their results (designated with a '*' in the results above). I then decided to see how the 4 common importance variables correlate to one another which can be seen in the pairwise scatterplot below:

Figure 1: Pairwise Scatterplot of Common Importance Variables



What I found interesting was that `area_worst` & `raduis_worst` were more correlated that `area_worst` & `area_mean`. What I initially thought before making `Figure 1` was that `area_worst` & `area_mean` would be more correlated than the other 2 variables as they are both under the same variable category.

4.2 area_mean

For `area_mean` after running the 3 regression models, I found the lowest `test error` is from the ridge regression model with an `RMSE` of `23.70795` with a `Rsquared` of `99.489%`. This mean that using ridge regression, we can explain `99.489%` of the variance in `area_mean` in Austin's and Danny's models. The ridge regression corresponding `training error` is `17.87197` with a `Rsquared` of `99.759%`. This corresponds to that `99.759%` of our variance in `area_mean` was trained in Austin's and Danny's models.

model	RMSE	Rsquared
Linear Train Error	17.61755	0.99766
Linear Test Error	25.23680	0.99437
Ridge Train Error	17.79893	0.99761
Ridge Test Error	23.99598	0.99476
Lasso Train Error	17.98211	0.99756
Lasso Test Error	24.05516	0.99474

4.3 Area_worst

In `area_worst` after running the 3 regression models, I found the lowest `test error` is from the ridge regression model with an `RMSE` of `42.28132` with a `Rsquared` of `99.288%`. This mean that using ridge regression, we can explain `99.288%` of the variance in `area_worst` in Austin's and Danny's models. The ridge

regression corresponding `training error` is `27.36780` with a `Rsquared` of `99.799%`. This corresponds to that `99.799%` of our variance in `area_worst` was trained in Austin's and Danny's models.

model	RMSE	Rsquared
Linear Train Error	26.50730	0.99811
Linear Test Error	43.78060	0.99300
Ridge Train Error	27.36780	0.99799
Ridge Test Error	42.28132	0.99288
Lasso Train Error	27.29435	0.99800
Lasso Test Error	44.28013	0.99219

4.4 concave.points_mean

With `concave.points_mean` after running the 3 regression models, I found the lowest `test error` is from the linear regression model with an `RMSE` of `0.00574` with a `Rsquared` of `97.847%`. This mean that using ridge regression, we can explain `97.847%` of the variance in `concave.points_mean` in Austin's and Danny's models. The ridge regression corresponding `training error` is `0.00467` with a `Rsquared` of `98.532%`. This corresponds to that `98.532%` of our variance in `concave.points_mean` was trained in Austin's and Danny's models.

model	RMSE	Rsquared
Linear Train Error	0.00467	0.98532
Linear Test Error	0.00574	0.97847
Ridge Train Error	0.00544	0.98008
Ridge Test Error	0.00603	0.97615
Lasso Train Error	0.00737	0.96348
Lasso Test Error	0.00777	0.96036

4.5 radius_worst

For `radius_worst` after running the 3 regression models, I found the lowest `test error` is from the lasso regression model with an `RMSE` of `0.24120` with a `Rsquared` of `99.713%`. This mean that using ridge regression, we can explain `99.713%` of the variance in `radius_worst` in Austin's and Danny's models. The ridge regression corresponding `training error` is `0.15907` with a `Rsquared` of `99.900%`. This corresponds to that `99.900%` of our variance in `radius_worst` was trained in Austin's and Danny's models.

model	RMSE	Rsquared
Linear Train Error	0.14130	0.99921
Linear Test Error	0.24134	0.99719
Ridge Train Error	0.17074	0.99885
Ridge Test Error	0.24127	0.99713
Lasso Train Error	0.15907	0.99900
Lasso Test Error	0.24120	0.99713

5 Analysis Summary

From the results of our classification analysis, we are confident not only in our predictions of tumor diagnosis, but reducing the amount of false-negative diagnoses. Through our methods of tuning specifically for Type II error, and lowering the prediction cutoff for malignant tumors, we have shown two successful ways of battling classification data with a preffered error class. This is extremely important in cancer diagnosis, as well as other analyses, as we do not want to tell someone they don't have cancer when they do, as that could delay treatment and further risk their life.

In terms of feature importance and feature selection, there were many limitations in interpretations due to the correlation of the measurement types. We did find that there were a number of the same features outlined by different models, so we can conclude that the variables, regardless of the measurement type, hold importance in classifying a tumor diagnosis.

6 Potential Improvements

Following our analysis, we have identified a few potential improvements that could be made to our analysis to further improve our models. The obvious and first improvement that could be made is to **collect more data**. With more data, we would allow our models to see more examples of malignant cells and be able to better classify them. As you saw in the Exploratory Data Analysis section, we have more more information regarding the benign cells than the malignant cells. If we had more data, we could balance out the data set and have more information regarding the malignant cells. Another possible improvement is to **collect different or more features** regarding each cell. Most of the features included the mean, worst and standard error of geometry measurements of the cell. This leads to many variables being highly correlated with each other and could lead to multicollinearity. If we had more features, we could reduce the multicollinearity and diversify the information we have regarding each cell. Finally, if we had **access to better technology** and more time or money, we could use a more advanced machine learning model. Because this project was done for a class, we were limited to the models we could use as there was a time contraint, especially in a team enviroment.

7 Works Cited

- Breast Cancer Wisconsin (Diagnostic) Data Set:
  - https://www.kaggle.com/datasets/uciml/breast-cancer-wisconsin-data?datasetId=180&sortBy=voteCount
- Definition of Features (Variables):
  - https://www.causeweb.org/usproc/sites/default/files/usclap/2017-2/Evaluating_Benign_and_Malignant_Breast_Cancer_Cells_from_Fine-Needle_Aspirates.pdf