Graph Analysis on Breast Cancer Data using Bayesian Networks Final Report

Kalea Sebesta[*]

University of Texas at San Antonio
k_sebesta@yahoo.com

July 19, 2018

Abstract

The purpose of this project is to create a Bayesian Belief Network (BBN) model to illustrate the relationships and impact of cancer nuclei measurements on the breast cancer diagnosis outcome. BBN are graphical mathematical models that use conditional probability distributions for the construction of the model.

Contents

[*] A special thank you to Dr. Han for the guidance and mentoring on this project

I. Introduction

Cancer is a terrifying diagnosis for any individual and along with fear, the diagnosis brings a number of uncertainties. The research to find answers for cancer patients is on the rise and special dedication to understanding the dynamics of the disease from a micro level are being conducted. For the purpose of this project, the scope will be narrowed to breast cancer and ten features that have been computed for each cell nucleus.

II. Methods & Experiments

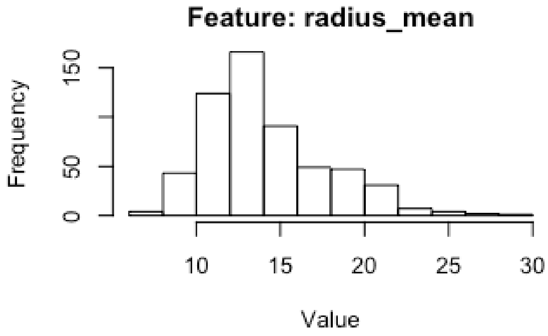
i. Data

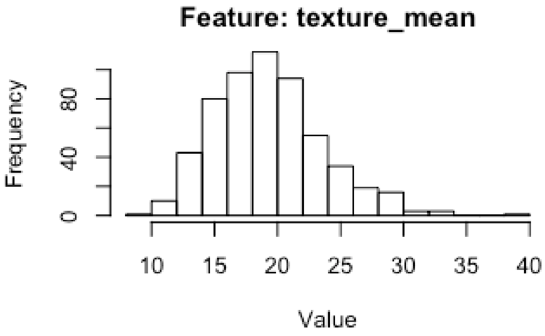
The data for this project can be found at Kaggle or on UCI Machine Learning Repository: https://www.kaggle.com/uciml/breast-cancer-wisconsin-data/data. The features in this dataset are originally computed from a digitized image of a fine needle aspirate (FNA) breast mass. This dataset contains the patient or image identification number, the diagnosis (malignant or benign), and ten specific features that have been calculated in three different ways (mean, standard error, and worst) thus resulting in 30 features. There are no missing values and the distribution of outcomes is 212 malignant and 357 benign. The ten cell nucleus features are:

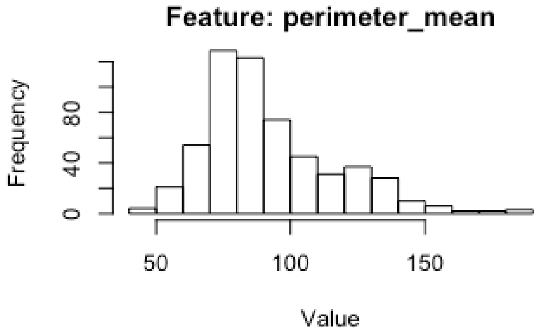
- Radius
- Texture
- Perimeter
- Area
- Smoothness
- Compactness
- Concavity
- Concave Points
- Symmetry
- Fractal Dimension

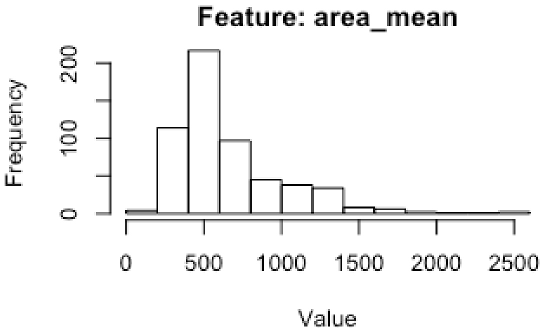
ii. Process

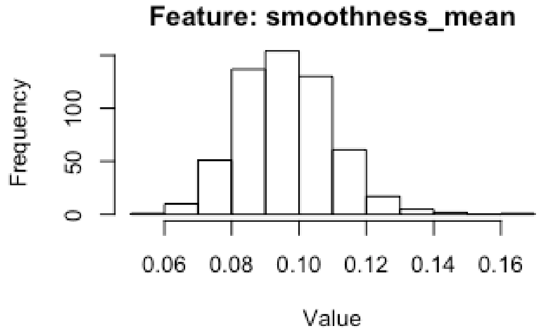
This project leveraged R programming language to preform and display the analytical and graphical results. As previously stated, the data set contains information on ten features for each cell nucleus of breast cancer patients. This data set was read into R and descriptive statistics were conducted to get a clearer picture of how the data looked. Of the 32 variables, there were 31 continuous variables and 1 discrete variable. 30 variables were features, 1 of the continuous variables was the patient id and held no true numerical value and thus was dropped from the set for analysis, the other 30 continuous variables were kept. The 1 discrete variable was the diagnosis variable which had two levels: B for benign and M for malignant. The breakdown for the diagnosis was 62.7% benign and 37.3% malignant. When reading the data in, R picked up that there was a variable called 'X' which had all missing values and thus this column was dropped from the data set for the analysis.

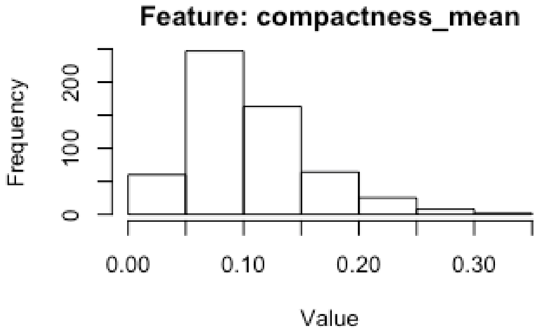

Feature: radius_mean
Frequency
0
50
150
10
15
20
25
30
Value

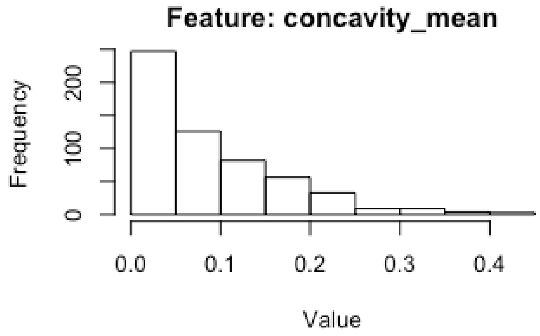

Feature: texture_mean
Frequency
0
40
80
10
15
20
25
30
35
40
Value

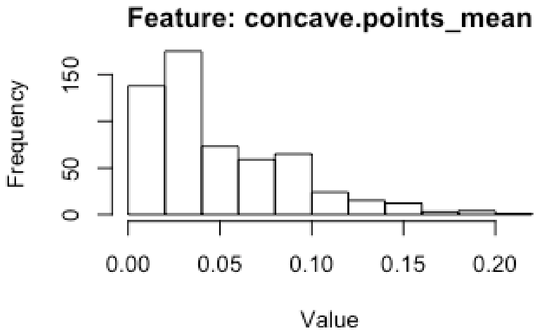

Feature: perimeter_mean
Frequency
0
40
80
50
100
150
Value

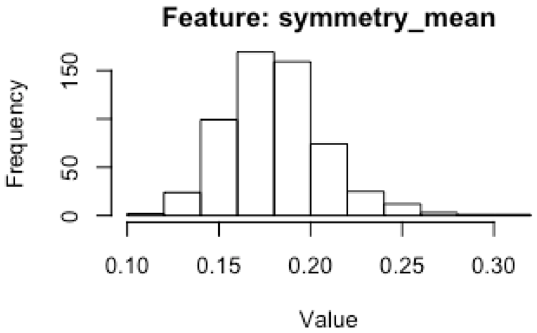

Feature: area_mean
Frequency
0
100
200
0
500
1000
1500
2000
2500
Value

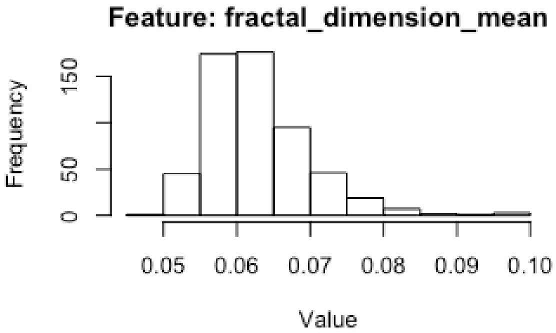

Feature: smoothness_mean
Frequency
0
50
100
0.06
0.08
0.10
0.12
0.14
0.16
Value

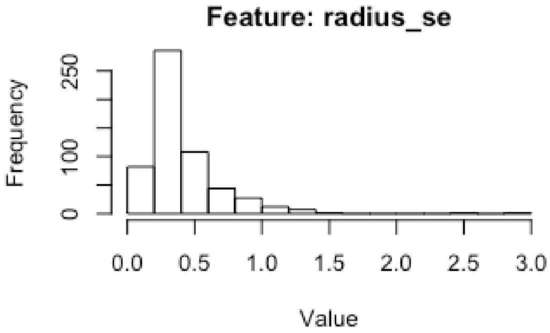

Feature: compactness_mean
Frequency
0
100
200
0.00
0.10
0.20
0.30
Value

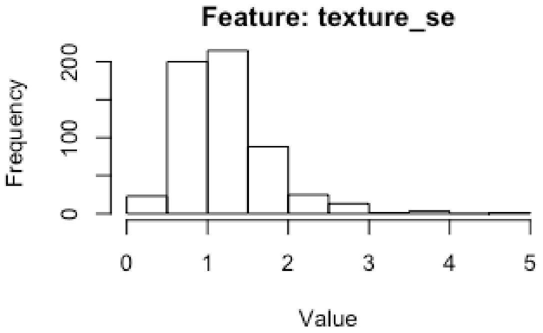

Feature: concavity_mean
Frequency
0
100
200
0.0
0.1
0.2
0.3
0.4
Value

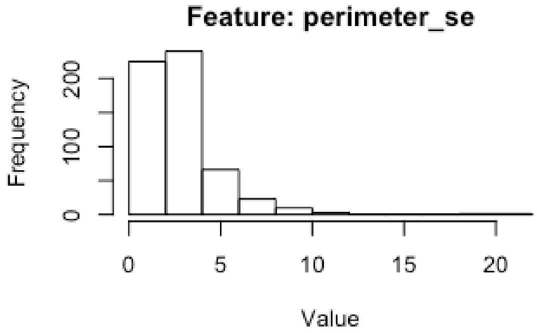

Feature: concave.points_mean
Frequency
0
50
150
0.00
0.05
0.10
0.15
0.20
Value

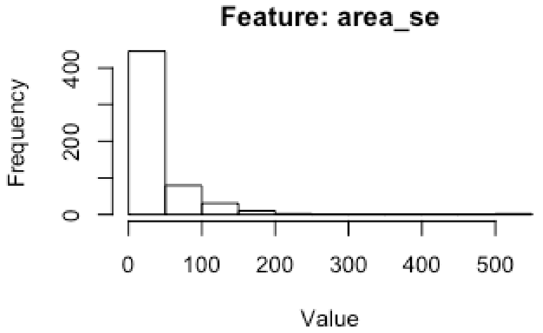

Feature: symmetry_mean
Frequency
0
50
150
0.10
0.15
0.20
0.25
0.30
Value

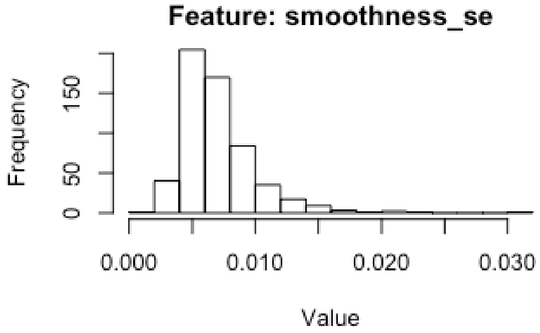

Feature: fractal_dimension_mean
Frequency
0
50
150
0.05
0.06
0.07
0.08
0.09
0.10
Value

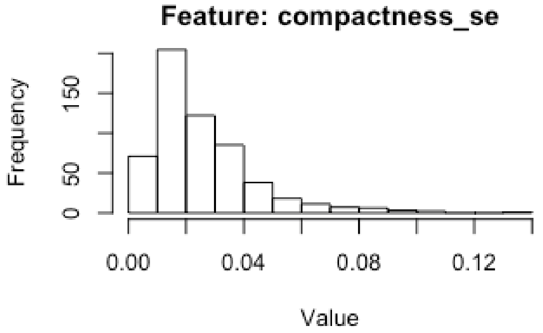

Feature: radius_se
Frequency
0
100
250
0.0
0.5
1.0
1.5
2.0
2.5
3.0
Value

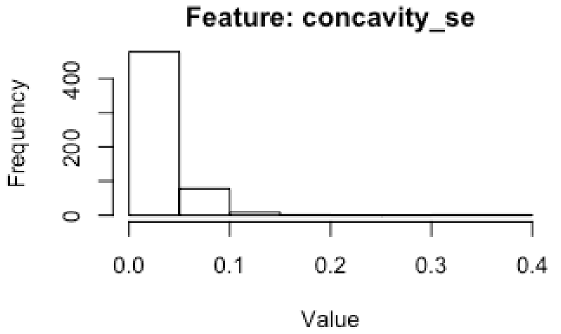

Feature: texture_se
Frequency
0
100
200
0
1
2
3
4
5
Value

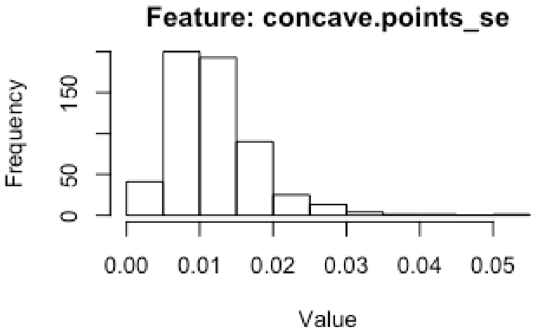

Feature: perimeter_se
Frequency
0
100
200
0
5
10
15
20
Value

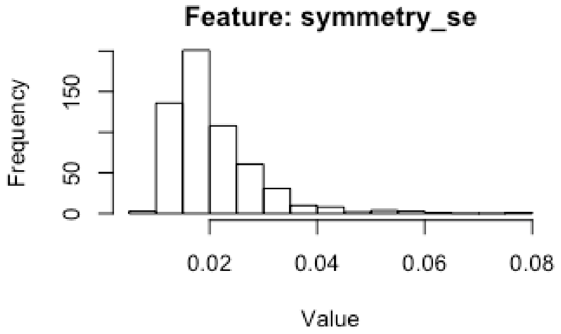

Feature: area_se
Frequency
0
200
400
0
100
200
300
400
500
Value

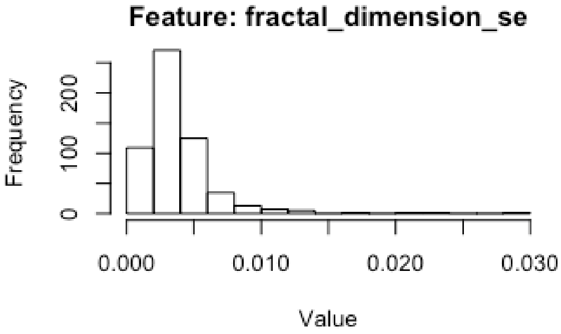

Feature: smoothness_se
Frequency
0
50
150
0.000
0.010
0.020
0.030
Value

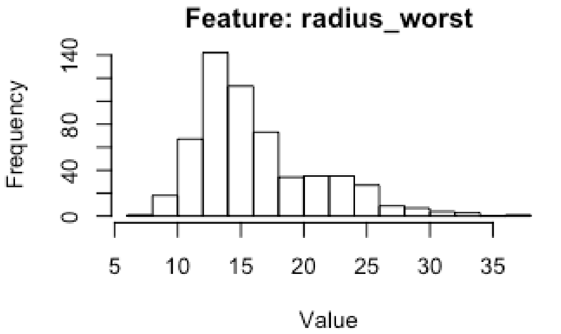

Feature: compactness_se
Frequency
0
50
150
0.00
0.04
0.08
0.12
Value

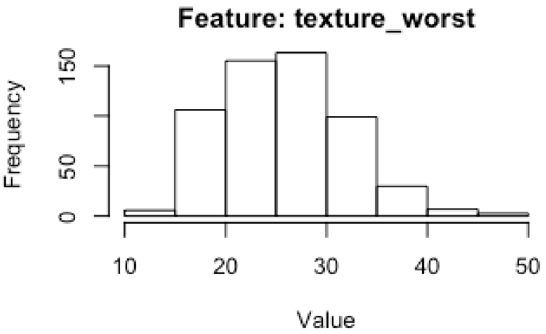

Feature: concavity_se

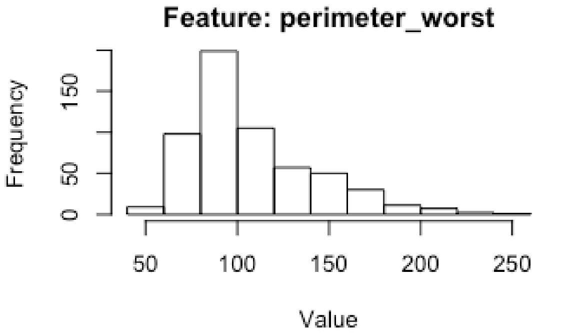

Feature: concave.points_se

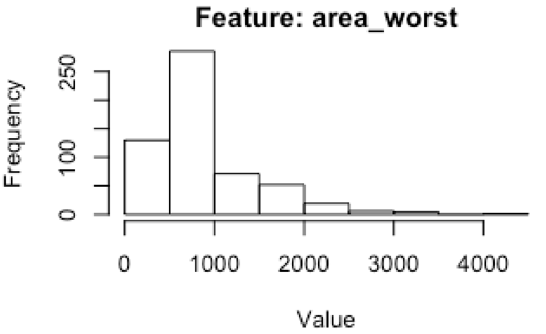

Feature: symmetry_se

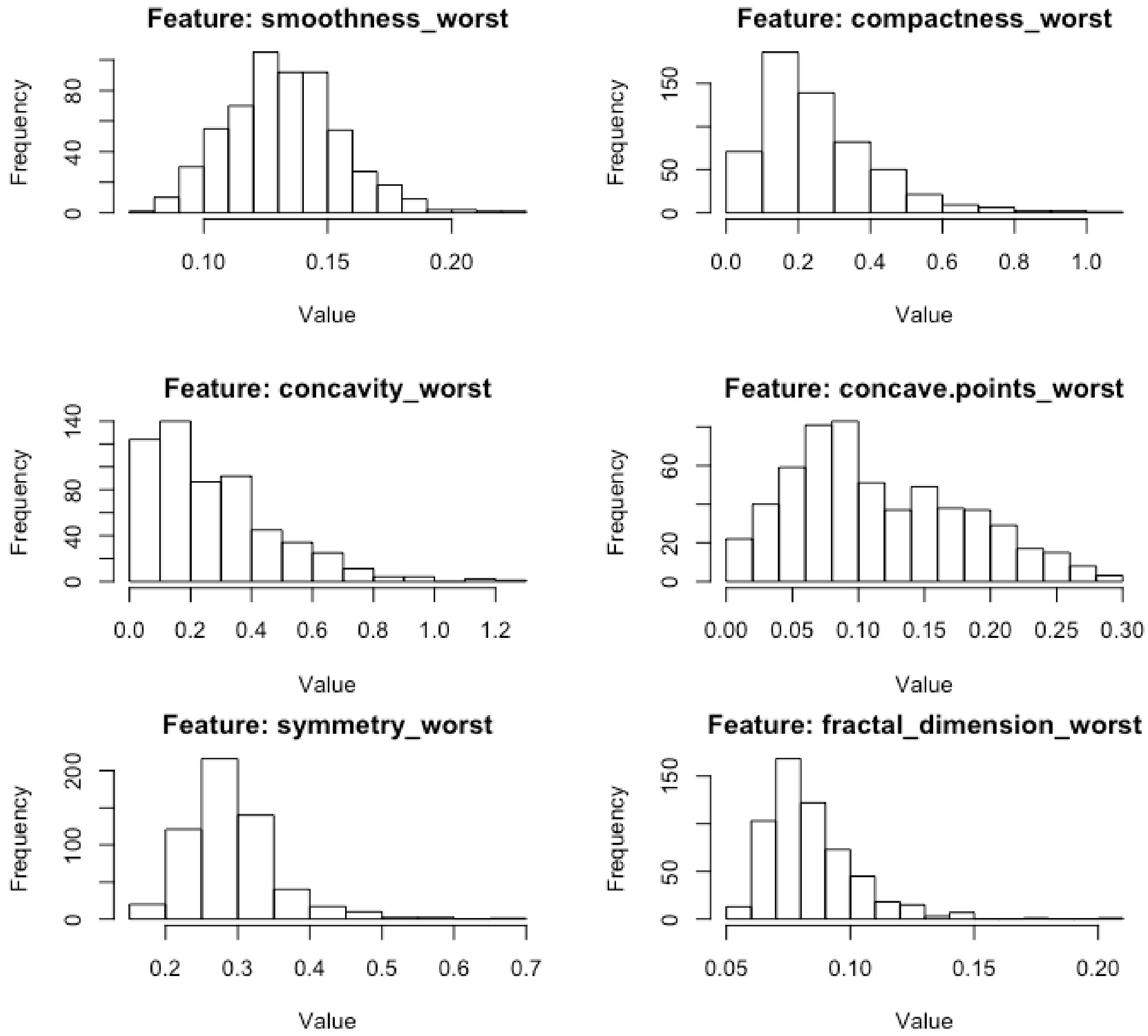

Feature: fractal_dimension_se


Feature: radius_worst


Feature: texture_worst

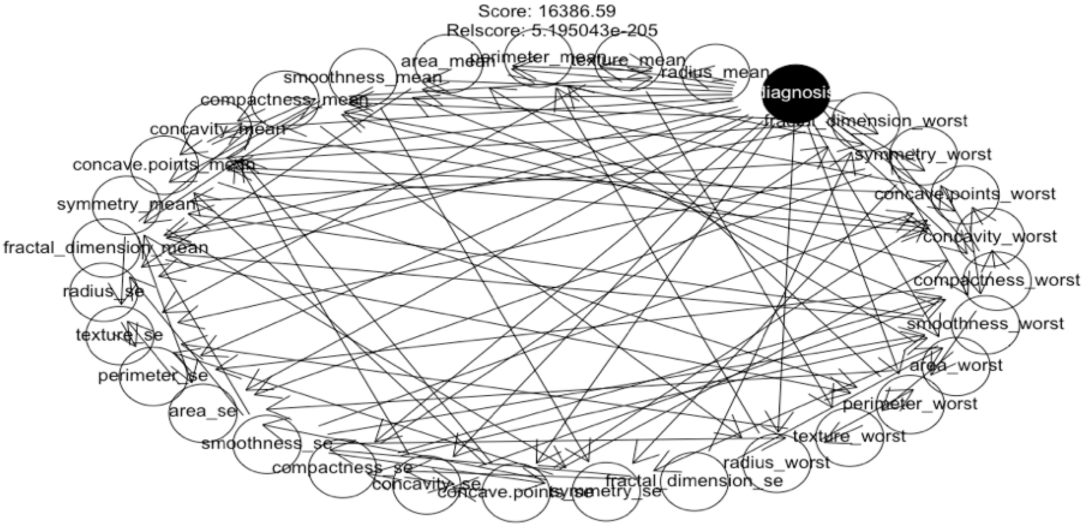

Feature: perimeter_worst


Feature: area_worst

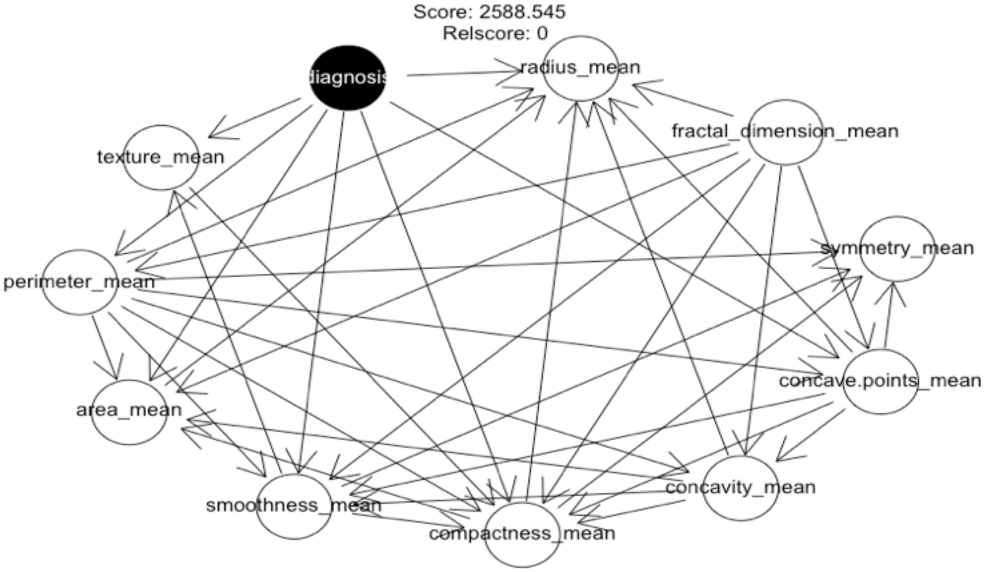


The DAGs created initially used all of the data set. Going forward, the data will need to be split into a training and test set and then the DAG can be reconstructed. The initial DAG incorporated all variables and then three additional DAGs were created with the mean variables, se variables, and worst variables. These structures were constructed by utilizing the learning structure functionality of the 'deal' package. The log network score was evaluated, learned, and updated for each node. The autosearch() and heuristic() functions were employed to learn the parameters of the network from the learned structure [4]. In the 'deal' package, the black nodes in the DAG are discrete variables and the white nodes are continuous variables. In the 'bnlearn' package all nodes are white. The first four DAGs were constructed with 'deal' and the last four DAGs were constructed with the 'bnlearn' package's score-based algorithm HC.

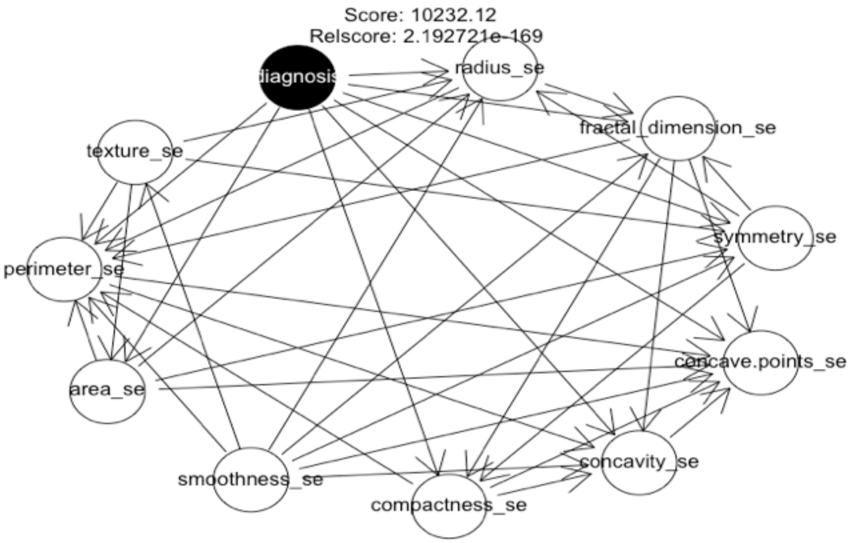
DAG Constructed on Full Data Set using 'deal'



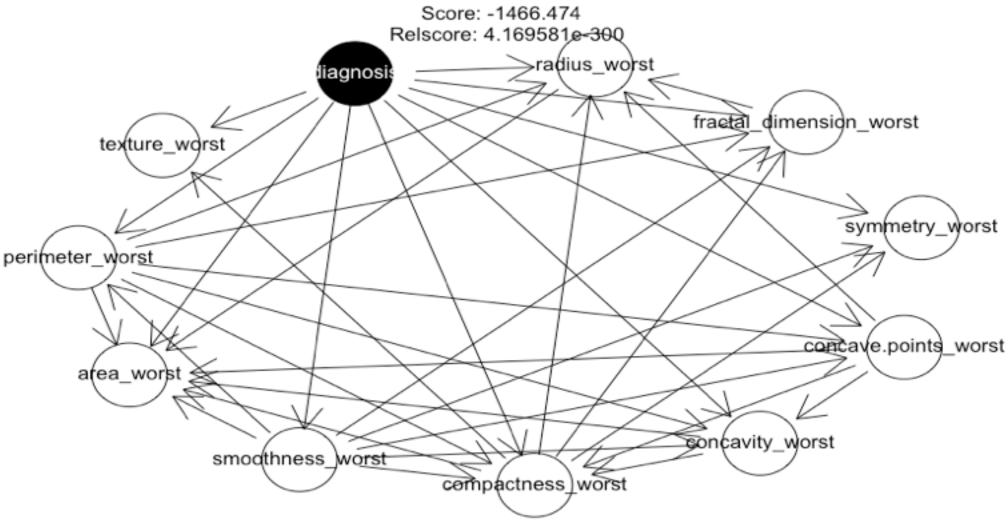
DAG Constructed on Mean Data Set using 'deal'



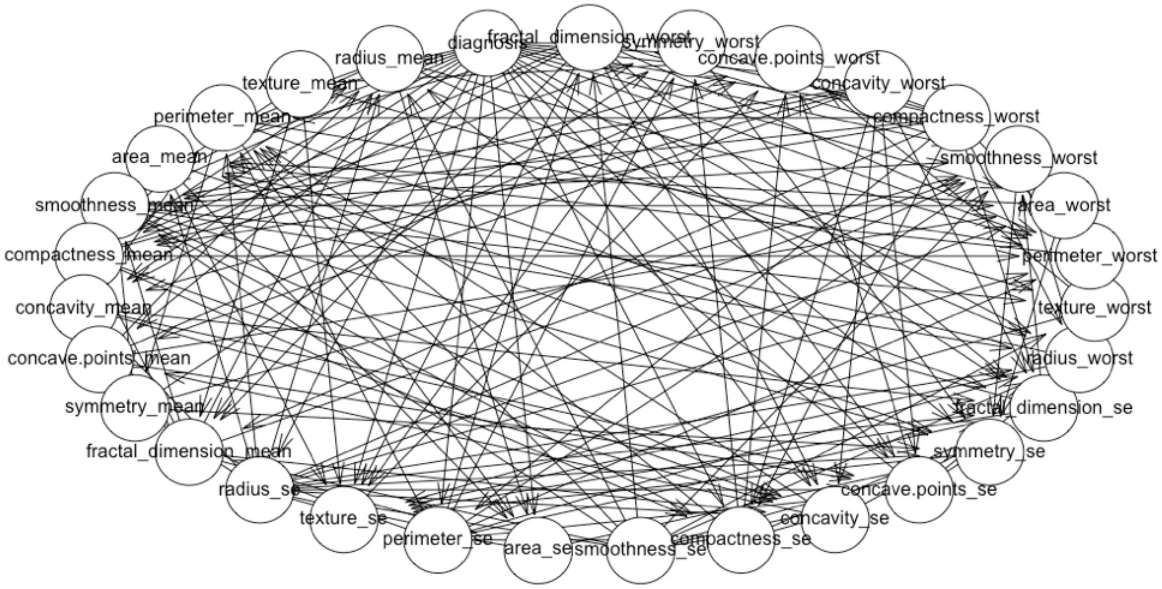
DAG Constructed on SE Data Set using 'deal'



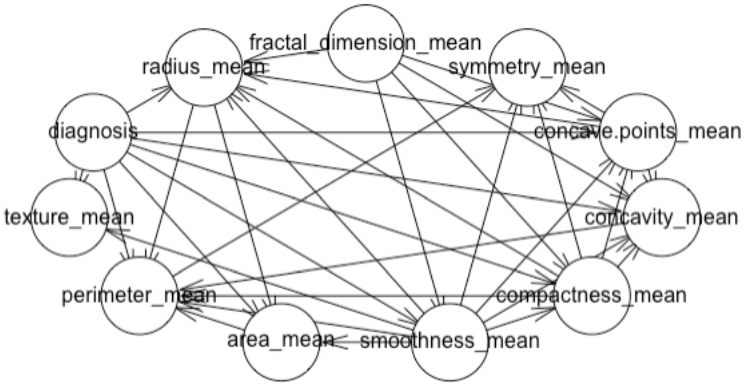
DAG Constructed on Worst Data Set using 'deal'



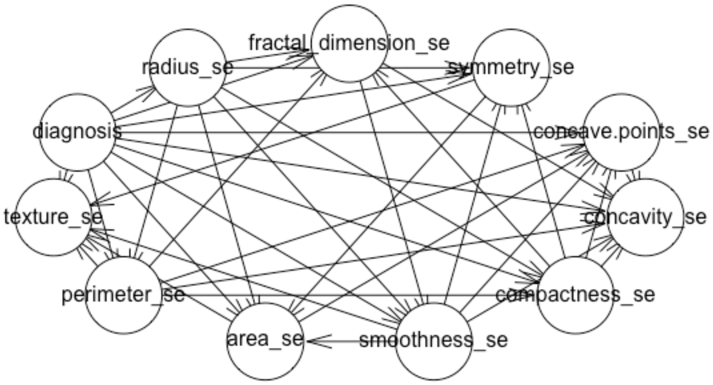
DAG Constructed on Full Data Set using 'bnlearn'



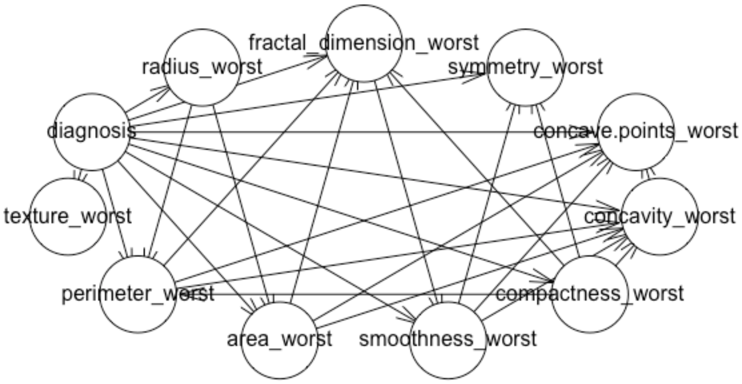
DAG Constructed on Mean Data Set using 'bnlearn'



DAG Constructed on SE Data Set using 'bnlearn'



DAG Constructed on Worst Data Set using 'bnlearn'



iii. Models

R can handle mixed variable types and use Gaussian distribution for the conditional probabilities. ‘bnlearn′ and ′klaR′ packages were the primary ones utilized to run the various models. Three modeling methods were used to construct bayesian models. The first model utilized the caret and klaR packages in R and the NaiveBayes algorithm. The dataset was split 80% for training and 20% for testing. Using all 30 predictors for the diagnosis outcome the following confusion matrix resulted:

	Actual	
Prediction	B	M
B	70	3
M	1	39

Table 1: *Naive Bayes Model One*

Recalling that the accuracy is calculated by the following formula and the corresponding variables live within the confusion matrix of the model.

$$a = \frac{TP + TN}{TP + TN + FP + FN}$$

In our particular situation the values for the variables are as follows

	Actual	
Prediction	B	M
B	TP	FN
M	FP	TN

Table 2: *Confusion Matrix Variable representation. TP represents the number of times that the prediction outcome was a true positive. This means that the prediction value and the actual value of the outcome was benign. FP represents a false positive meaning that the prediction outcome was malignant when the actual outcome was benign. Likewise, FN means a benign prediction outcome occurred when the actual outcome was malignant and a TN is a true negative meaning the prediction and actual outcome were malignant.*

$$TP = \text{True Positive} = 70$$

$$TN = \text{True Negative} = 39$$

$$FP = \text{False Positive} = 1$$

$$FN = \text{False Negative} = 3$$

Substituting these values into the formula, the accuracy is calculated:

$$a = \frac{TP + TN}{TP + TN + FP + FN}$$

$$a = \frac{70 + 39}{70 + 39 + 1 + 3}$$

$$a = \frac{109}{113} = 96.46\%$$

Other values for this model are as follows:

Accuracy	0.9646
95% CI	(0.9118, 0.9903)
No Information Rate	0.6283
P-Value [Acc > NIR]	<2e-16
Kappa	0.9235
Mcnemar's Test P-Value	0.6171
Sensitivity	0.9859
Specificity	0.9286
Pos Pred Value	0.9589
Neg Pred Value	0.9750
Prevalence	0.6283
Detection Rate	0.6195
Detection Prevalence	0.6460
Balanced Accuracy	0.9572

Table 3: *Outcome Results for Model One*

The second model utilized the boot strapping technique. In this model the resampling had a cross validation of 10 fold, repeated 3 times. The tuning parameter 'fL' was held constant at zero and the 'adjust' parameter was held at a constant of one. The resampling results across the tuning parameters are as follows:

usekernel	Accuracy	Kappa
False	0.9325584	0.8548093
True	0.9419562	0.8758492

Table 4: *Model Two Outcome: Bootstrap resampling method with 10 fold cross-validation repeated 3 times*

The third model utilized the Leave Out One Cross Validation technique. The tuning parameters were set as 'fL' was held constant at a value of zero and 'adjust' was held constant at a value of one. Accuracy was used to select the optimal model using the largest value. The final values used for the model were fL = 0, usekernel = TRUE and adjust = 1. Resampling results across tuning parameters resulted in the following:

usekernel	Accuracy	Kappa
False	0.9332162	0.8563303
True	0.9420035	0.8758324

Table 5: *Model Three Outcome: Leave-One-Out Cross-Validation*

III. Results & Timeline

The three models are compared looking at the average accuracy for each model. It is important to note that these models are without applying the synthetic minor oversampling technique known as SMOTE.

Model	Accuracy	Kappa
Model 1 (NaiveBayes)	96.46%	92.35%
Model 2 (Bootstrap)	93.72%	86.53%
Model 3 (LOOCV)	93.76%	86.61%

Table 6: *Comparison result across models*

Table 7: *Milestone Timeline*

Expectations		
Date	Milestone	Status Report Week
May 25	Preliminary Research on Past Work	1 Completed
May 25	Data Exploration/ Preliminary Analysis	1 Completed
June 1	Creating Bayesian Model using naiveBayes Approach	2 Completed
June 1	Compare Models	2 Completed
June 8	Use SMOTE to balance dataset then rerun models	3 Completed
June 8	Research physiochemical understanding of the oncogenic process	3 In Progress
June 8	Create Models with Subset Feature Selection	3 Completed
June 8	Create Density Plots	3Completed
June 15	Use SMOTE to balance dataset then rerun models	4 Completed
June 16	Research optimization techniques to increase model performance	4 Completed
June 22	Apply optimization techniques to increase model performance	4 In Progress
June 29	(Discretize Variables for Optimization) Parameter Learning	6 Completed
July 27	Final Report	10 Completed

IV. Conclusion

After the initial models were ran, the SMOTE method was used to sample the minority class and balance the dataset. Along with SMOTE, the data was also discretized. The median of each feature was used as the cut off point. For values less than the medium, they were converted to zero and values for greater than the medium were converted to one. The goal of SMOTE and discretizing the variables was to rerun the model, tune the parameters and ultimately optimize the already well performing model. Due to time constraints this portion of the project was unable to be completed.

V. Challenges & Next Steps

The biggest challenge during this project was finding expert opinion regarding the relationships between the variables in the dataset to determine if arcs could be removed from the DAG. Another challenge that occurred was utilizing the 'awnb' functionality which would allow for parameter tuning to occur. It appeared that after discretizing the variables there was still an issue with factorizing the variables to the form that 'awnb' allowed. The next steps for this project could include but are not limited to the following:

1. Factorize variables for the 'awnb' function
2. Implement 'awnb' for model performance increase

References

[1] Taheri, S., Yearwood, J., Mammadov, M., and Seifollahi, S., (2014). Attribute weighted Naive Bayes classifier using a local optimization *Springer*

[2] Nagarajan, R., Scutari, M., and Lebre, S., (2013). Bayesian Networks in R with Applications in Systems Biology *Springer (US)*, Vol. 48.

[3] Nagarajan, Scutari, and Lebre, 2013 Learning Bayesian Networks with the bnlearn R Packages *Journal of Statistical Software*, Vol. VV., Issue II Scutari, M., (2010).

[4] Bottcher, S, Dethlefsen, C, 2003 deal: A Package for Learning Bayesian Network *Journal of Statistical Software*, Vol. 8, Issue 20

[5] Scutari, M. and Denis, J. B., (2014). Bayesian Networks in R with Examples in R *Texts in Statistical Science, Chapman & Hall/CRC (US)*.