STP 494 Final

Jiaqi Wu, Excel Ortega, Terry Wen, Alex Ryan, Peter Le

April 22, 2018

Introduction

This project will attempt to classify breast cancer tumors into two categories, benign or malignant, depending on tumor characteristics.

Data

The Wisconsin Diagnostic Breast Cancer (WDBC) dataset was obtained from Kaggle. Attributes include ID number, diagnosis (B = benign, M = malignant), and 30 real-valued input features. This project will focus on 11/32 attributes given in this dataset.

Features are computed from a digitized image of breast mass, and the data describes characteristics of the cell nuclei present in the image.

Variable Name:	Variable Description:
`radius`	Mean of distances from center to points on the perimeter
`texture`	Standard deviation of gray-scale values
`perimeter`	Measurement of cell boundary
`area`	Size of the cell's 2D surface
`smoothness`	Local variation in radius lengths
`compactness`	Perimeter^2 / Area - 1.0
`concavity`	Severity of concave portions of the contour
`concave points`	Number of concave portions of the contour
`symmetry`	Similarity of parts around an axis
`fractal dimension`	"Coastline approximation" - 1

The mean, standard error, and "worst" (largest) of each of the 10 features were computed, resulting in 30 attributes total. This project only looks at the "mean" values.

Processing the Data

First, we removed the ID attribute from the dataset because it does not correlate with diagnosis. Then, we created a dataset with only the variables we wantd to use: columns 1 through 11 (diagnosis column and 10 mean-value features).

We also added a column that maps B (benign) to 0 and M (malignant) to 1. This column contains the same information as the original diagnosis column, except the values are 0 and 1 rather than B and M (respectively). In total, the dataset contains 569 observations and 12 variables. We split the data into two sets to build our models: 70% training data, and 30% test data.

Data Models

We attempted four methods from our Machine Learning class: logistic regression, k-nearest neighbors, naive bayesian classifier, and neural nets. Logistic regression was our baseline reference, but the remaining models

were tuned for optimal performance.

Logistic Regression

We used generalized linear models (glm) to model our data. Since the outcome we are trying to predict is binomial (benign or malignant), we chose a logistic regression model, given below:

```
##
## Call:
## glm(formula = diagnosis2_train ~ ., data = wdbc_train[-c(1)])
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -0.69561  -0.18203  -0.03698   0.17591   0.83032
##
## Coefficients:
##                           Estimate Std. Error t value Pr(>|t|)
## (Intercept)            -2.1978556  0.5048502  -4.353 1.72e-05 ***
## radius_mean             0.5575800  0.1529817   3.645 0.000304 ***
## texture_mean            0.0198284  0.0034554   5.738 1.93e-08 ***
## perimeter_mean         -0.0639490  0.0245334  -2.607 0.009498 **
## area_mean              -0.0009843  0.0002875  -3.424 0.000682 ***
## smoothness_mean         3.0634844  1.6965888   1.806 0.071746 .
## compactness_mean       -0.0489724  1.2760931  -0.038 0.969407
## concavity_mean          1.4742695  0.6349484   2.322 0.020758 *
## concave.points_mean     5.0578622  1.6551181   3.056 0.002399 **
## symmetry_mean           1.0367515  0.6717178   1.543 0.123543
## fractal_dimension_mean -0.2618018  5.0987642  -0.051 0.959076
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for gaussian family taken to be 0.07731082)
##
##     Null deviance: 95.068  on 397  degrees of freedom
## Residual deviance: 29.919  on 387  degrees of freedom
## AIC: 123.47
##
## Number of Fisher Scoring iterations: 2
```

Confusion Matrix:

```
##        wdbc_testy
## yhatbin   0   1
##       0 111   8
##       1   5  47
```

This model gives an accuracy of 92.4%.

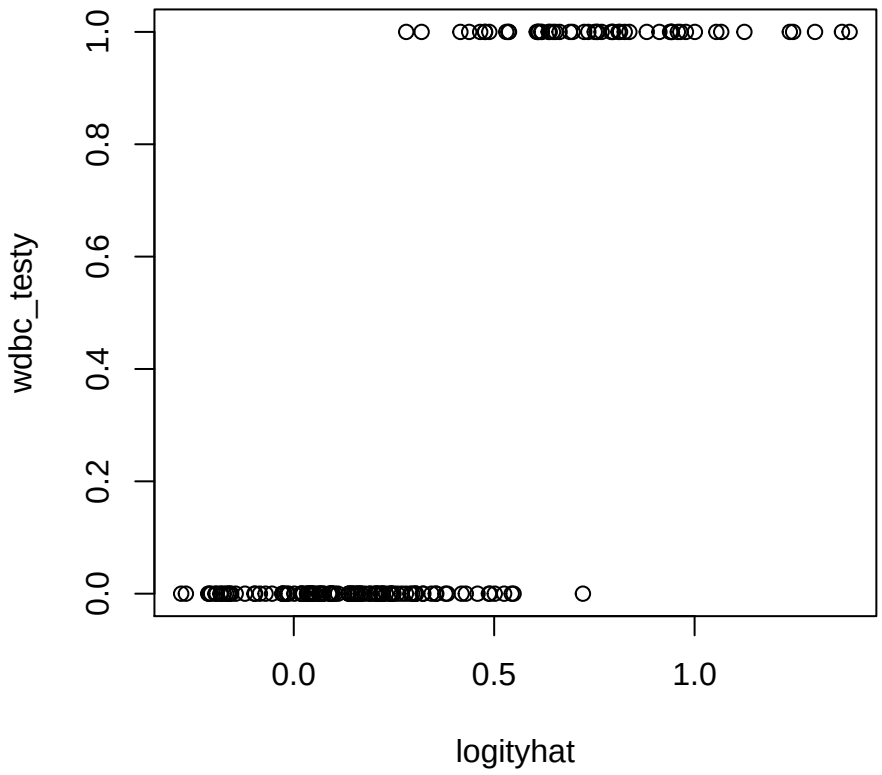


Figure 1.: Shows the probabilities of diagnosis predicted by our logistic model, versus the actual diagnosis. When computing accuracy, values greater than 0.5 were assigned a value of 1 (malignant), while values less than or equal to 0.5 were assigned a value of 0 (benign).

kNN

We used the r package **kknn** to fit our knn model. To begin, we ran cross-validation once with a large interval of k-values to find an appropriate range of values. Then, with a smaller interval, we ran 10-fold cross validation multiple times to find the optimal k-value. We fit our test data using the optimal k-value, and generated plots and a confusion matrix to determine accuracy.

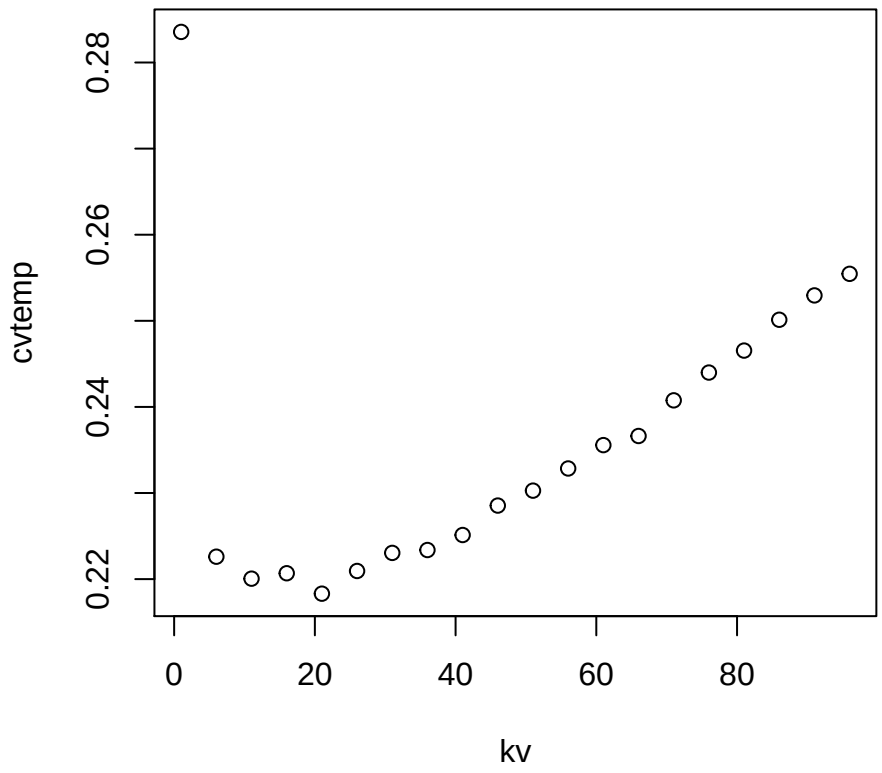


Figure 2. Comparing the k parameter from 1 to 100 in intervals of 5 with the RMSE of a single model. This allows us to approximate range of appropriate k-values before moving onto a more computationally intensive model.

It looks like the optimal k-value is somewhere around k = 19, so we will reset the kv-value to a smaller window (2 to 30) for 10-fold cross validation.

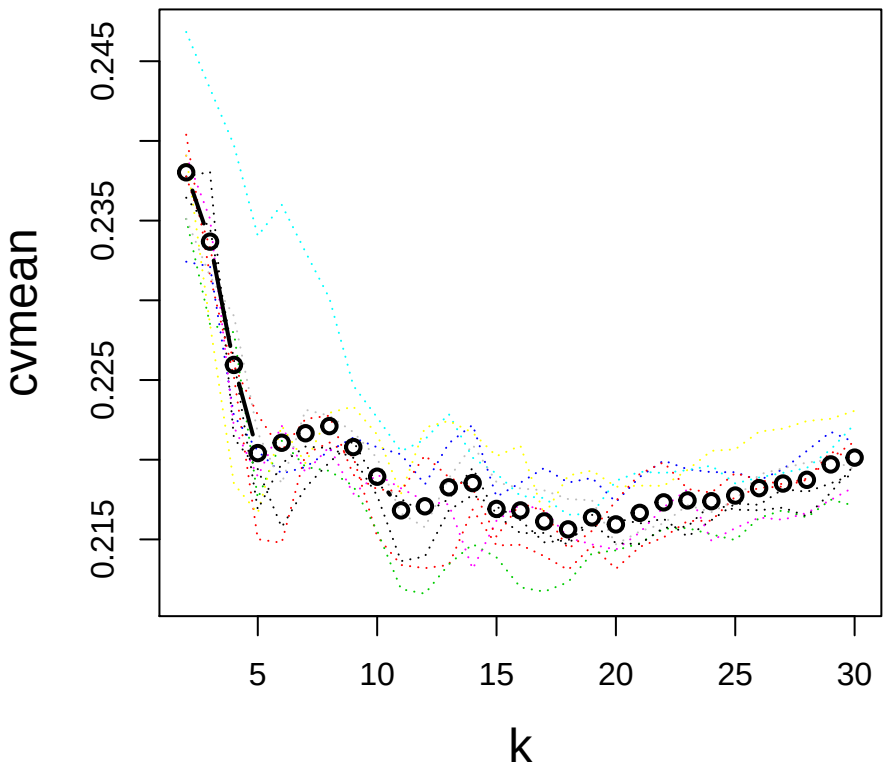


Figure 3. This plot compares the paramter k at values between 2 and 30 with the RMSE of each model. Each colored line represents a different subset of the data; the black line represents the average values. This helps us find the balance between bias and variance for the model.

```
## k= 2  rmse= 0.23368
## k= 3  rmse= 0.2259472
## k= 4  rmse= 0.2204153
## k= 5  rmse= 0.2210641
## k= 6  rmse= 0.2216606
## k= 7  rmse= 0.2221094
## k= 8  rmse= 0.2207938
## k= 9  rmse= 0.2189341
## k= 10  rmse= 0.2168223
## k= 11  rmse= 0.2170778
## k= 12  rmse= 0.2182672
## k= 13  rmse= 0.2185329
## k= 14  rmse= 0.216912
## k= 15  rmse= 0.2168212
## k= 16  rmse= 0.2161294
## k= 17  rmse= 0.2156284
## k= 18  rmse= 0.2163824
## k= 19  rmse= 0.2159413
## k= 20  rmse= 0.2166654
## k= 21  rmse= 0.2173233
```

```
## k= 22  rmse= 0.2174459
## k= 23  rmse= 0.2173951
## k= 24  rmse= 0.2177519
## k= 25  rmse= 0.2182153
## k= 26  rmse= 0.2185053
## k= 27  rmse= 0.2187396
## k= 28  rmse= 0.2197041
## k= 29  rmse= 0.2201308
```

Our optimal k-value is 17 with an average RMSE of 0.2156284. This is pretty close to our initial approximation. Let's refit using k = 17:

```
##
##       B   M
##   B 113   5
##   M   3  50
```

The confusion matrix shows that kNN performs decently well, with 95.32% accuracy.

```
plot(near17$prob[,2], wdbc_testy)
```

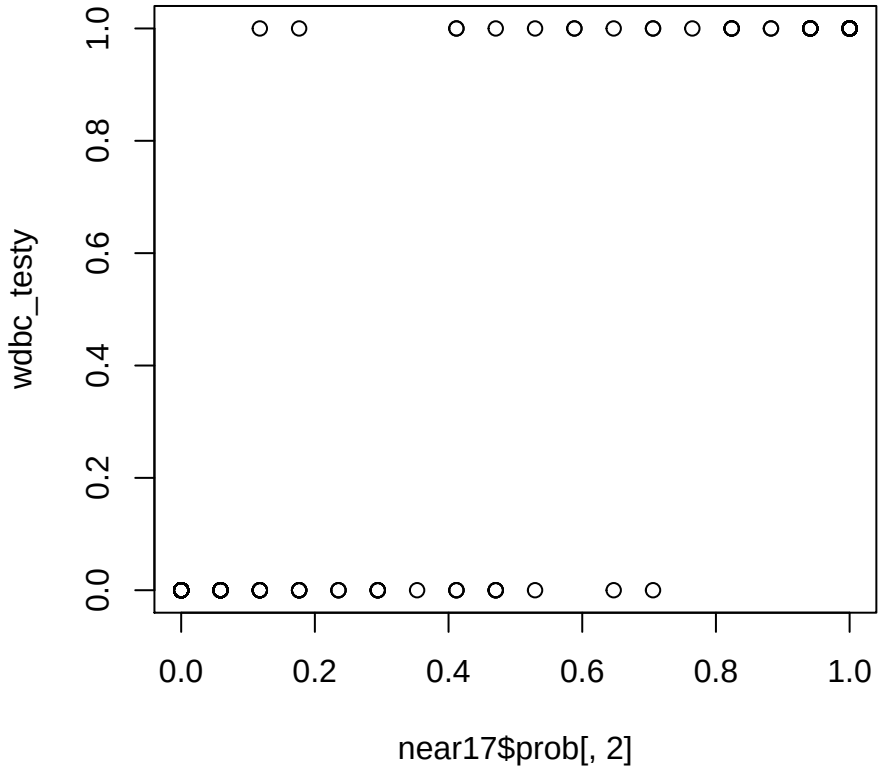


Figure 4. Probabilities predicted by our kNN model versus actual diagnosis.

Naive Bayesian Classifier

We used a Naive Bayesian Classifier (e1071) to model the data on on a conditionally independent basis. The outcome prediction is binomial, which is a great fit for Naive Bayesian Classifiers. After tuning the laplace value for lowest misclassification rate, we ran Naive Bayes classification with laplace=1.

Confusion Matrix:

```
##       wdbc_testy
## nbHat   B   M
##     B 111   2
##     M   5  53
```

Our Naive Bayesian classifer model has an accuracy of 95.91%.

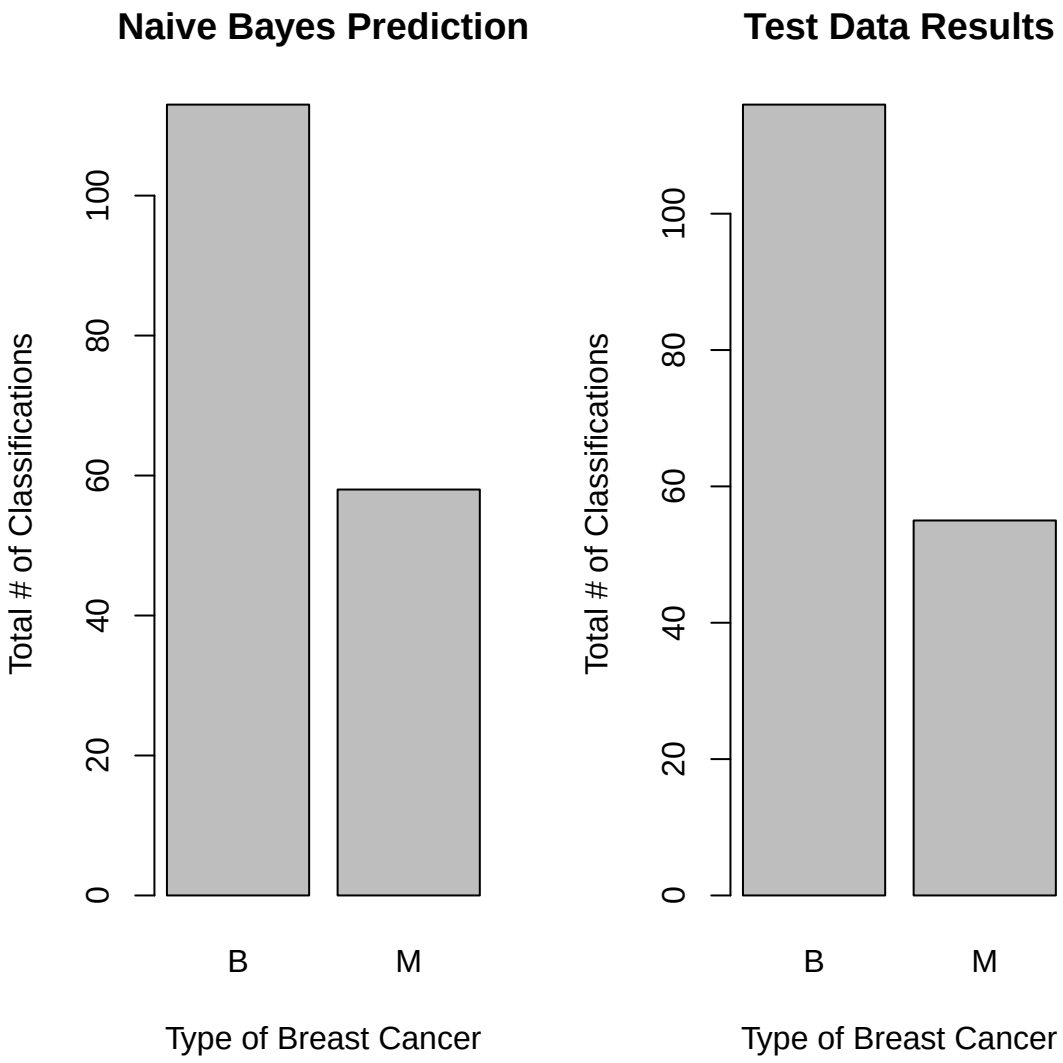


Figure 5: A side by side comparison of Naive Bayes Classifier predictions and actual test data classifications.

Neural Nets

We used a neural network (nnet) package to create a single layer neural net. Tuning was performed with multiple network structures, with different numbers of nodes in the hidden layer. After tuning and analysis, our chosen model included a hidden layer with 5 nodes.

```
## Warning: package 'nnet' was built under R version 3.4.4
```

```
## # weights:  61
## initial  value 285.313668
## iter  10 value 266.750392
## iter  20 value 196.516971
## iter  30 value 146.540847
## iter  40 value 140.648251
## iter  50 value 120.400112
## iter  60 value 87.688263
## iter  70 value 73.910000
## iter  80 value 73.185445
## iter  90 value 73.107601
## iter 100 value 73.106078
## iter 110 value 73.105439
## iter 110 value 73.105438
## iter 110 value 73.105438
## final  value 73.105438
## converged
```

```
##           wdbc_testy
## nnyhat_bin   B   M
##          0 111   4
##          1   5  51
```

Our neural nets model has an accuracy of 94.74%. While this percentage is still decent, we are concerned because the errors are more widespread (i.e., incorrectly predicted malignant points have values very close to 0, and vice versa).

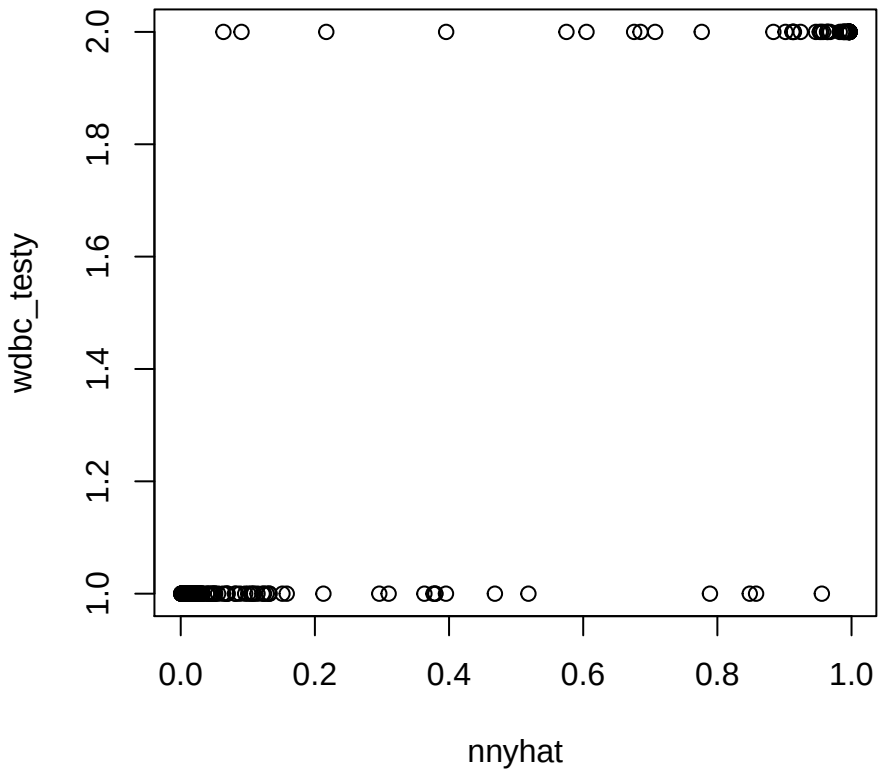
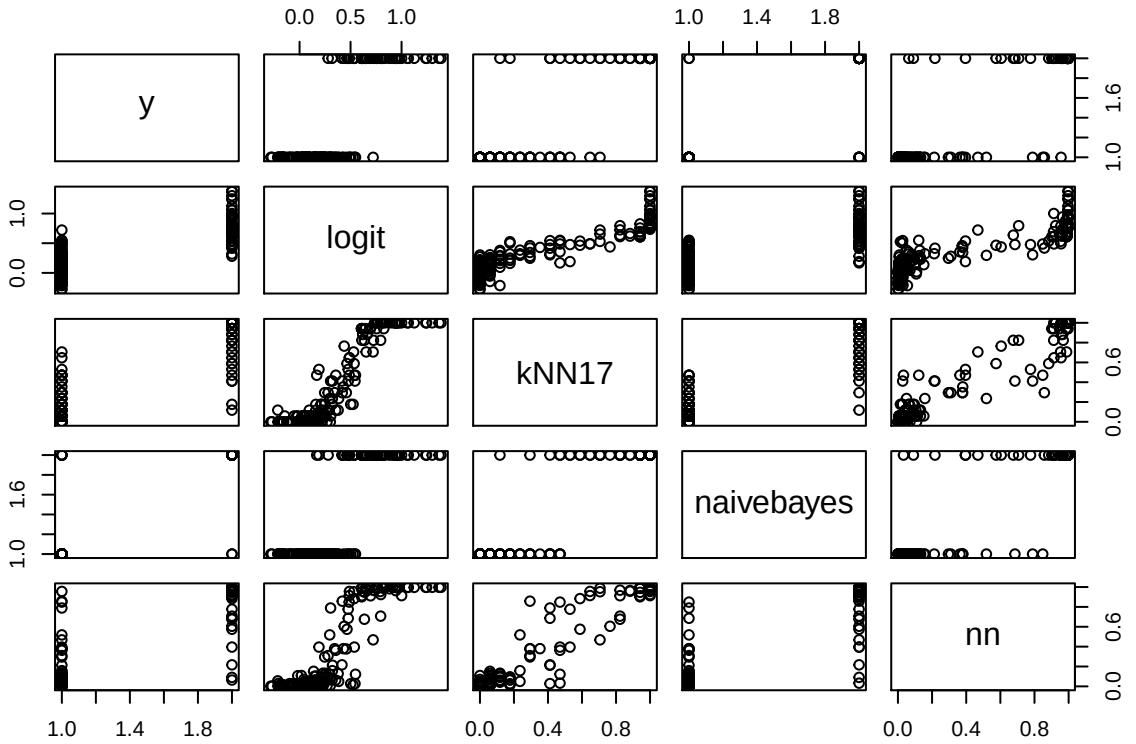


Figure 6. Probabilities predicted by neural nets model versus actual diagnosis.

Conclusion




```
##                    y     logit     kNN17 naivebayes        nn
## y          1.0000000 0.8159301 0.9001624  0.9082313 0.8859457
## logit      0.8159301 1.0000000 0.9115989  0.8115188 0.8742495
## kNN17      0.9001624 0.9115989 1.0000000  0.9199657 0.9616809
## naivebayes 0.9082313 0.8115188 0.9199657  1.0000000 0.9013086
## nn         0.8859457 0.8742495 0.9616809  0.9013086 1.0000000
```

Figure 7. All models versus real diagnosis values.

Based on the accuracy of each model, we can compare each model and rank their effectiveness. Each of the accuracy was computed from the generate confusion matrix. Our Naive-Bayes classifier has the highest accuracy with 95.91%, followed by KNN at 95.32% accuracy and neural nets at 94.74%. These three models all performed better than logistic regression at 92.40%. Based on the correlation table generated above, the correlations also rank the models in the same order, with logistic regression having the smallest correlation. In future work, these models could be adjusted more, tuning parameters with a separate set of cross-validation data to improve upon test performance.