Comparative Study of classification techniques using Breast Cancer Wisconsin Diagnostic Data Set

COM737 Machine Learning and Data Modelling - 2018/19 Semester 2

B00749292 – Surya Patnaik

Outline

Project Proposal

Project Plan

Exploratory Data Analysis

Machine Learning

Strengths & Limitations

Results/Conclusion

Project Proposal

The proposal is to analyze the Wisconsin Breast Cancer Diagnostic dataset and apply multiple machine learning techniques and compare the performance of the implemented machine learning techniques.

This dataset is publicly available from the UCI Machine Learning Data Repository. Following is a high level project plan.

Work package	01-Apr-19	03-Apr-19	08-Apr-19	15-Apr-19	22-Apr-19	28-Apr-19
Project Proposal						
Data Acquisition						
Exploratory Data Analysis						
Model Development						
Paper preparation & Review						
Submission						

Risk Management

There are no health and safety risks associated with this project.

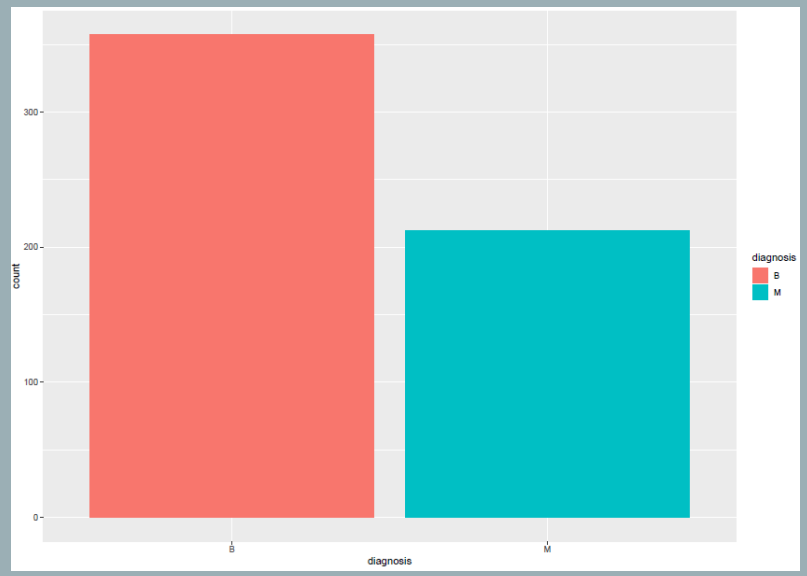
Risk	Type of Risk	Probability	Loss	Threat (probability * loss)	Remedial Action
Volume of data	Technical Risk	2	4	8	Select subset of devices
Hardware Limitation	Technical Risk	2	3	6	Upgrade RAM
Domain Knowledge	Technical Risk	2.5	3	7.5	More research required
Software Limitation	Technical Risk	2	3	6	Try other software's or Algorithms
Not completing on time	Management Risk	1.5	4	6	Contingency planning

Exploratory Data Analysis

Structure of dataset

```
> str(wbcd)
'data.frame':   569 obs. of  32 variables:
 $ id               : int  87139402 8910251 905520 868871 9012568 906539 925291 87880 862989 89827 ...
 $ diagnosis        : Factor w/ 2 levels "B","M": 1 1 1 1 1 1 1 1 2 1 1 ...
 $ radius_mean      : num  12.3 10.6 11 11.3 15.2 ...
 $ texture_mean     : num  12.4 18.9 16.8 13.4 13.2 ...
 $ perimeter_mean   : num  78.8 69.3 70.9 73 97.7 ...
 $ area_mean        : num  464 346 373 385 712 ...
 $ smoothness_mean  : num  0.1028 0.0969 0.1077 0.1164 0.0796 ...
 $ compactness_mean : num  0.0698 0.1147 0.078 0.1136 0.0693 ...
 $ concavity_mean   : num  0.0399 0.0639 0.0305 0.0464 0.0339 ...
 $ points_mean      : num  0.037 0.0264 0.0248 0.048 0.0266 ...
 $ symmetry_mean    : num  0.196 0.192 0.171 0.177 0.172 ...
 $ dimension_mean   : num  0.0595 0.0649 0.0634 0.0607 0.0554 ...
 $ radius_se        : num  0.236 0.451 0.197 0.338 0.178 ...
 $ texture_se       : num  0.666 1.197 1.387 1.343 0.412 ...
 $ perimeter_se     : num  1.67 3.43 1.34 1.85 1.34 ...
 $ area_se          : num  17.4 27.1 13.5 26.3 17.7 ...
 $ smoothness_se    : num  0.00805 0.00747 0.00516 0.01127 0.00501 ...
 $ compactness_se   : num  0.0118 0.03581 0.00936 0.03498 0.01485 ...
 $ concavity_se     : num  0.0168 0.0335 0.0106 0.0219 0.0155 ...
 $ points_se        : num  0.01241 0.01365 0.00748 0.01965 0.00915 ...
 $ symmetry_se      : num  0.0192 0.035 0.0172 0.0158 0.0165 ...
 $ dimension_se     : num  0.00225 0.00332 0.0022 0.00344 0.00177 ...
 $ radius_worst     : num  13.5 11.9 12.4 11.9 16.2 ...
 $ texture_worst    : num  15.6 22.9 26.4 15.8 15.7 ...
 $ perimeter_worst  : num  87 78.3 79.9 76.5 104.5 ...
 $ area_worst       : num  549 425 471 434 819 ...
 $ smoothness_worst : num  0.139 0.121 0.137 0.137 0.113 ...
 $ compactness_worst: num  0.127 0.252 0.148 0.182 0.174 ...
 $ concavity_worst  : num  0.1242 0.1916 0.1067 0.0867 0.1362 ...
 $ points_worst     : num  0.0939 0.0793 0.0743 0.0861 0.0818 ...
 $ symmetry_worst   : num  0.283 0.294 0.3 0.21 0.249 ...
 $ dimension_worst  : num  0.0677 0.0759 0.0788 0.0678 0.0677 ...
```

Diagnosis split
357 Benign
212 Malignant

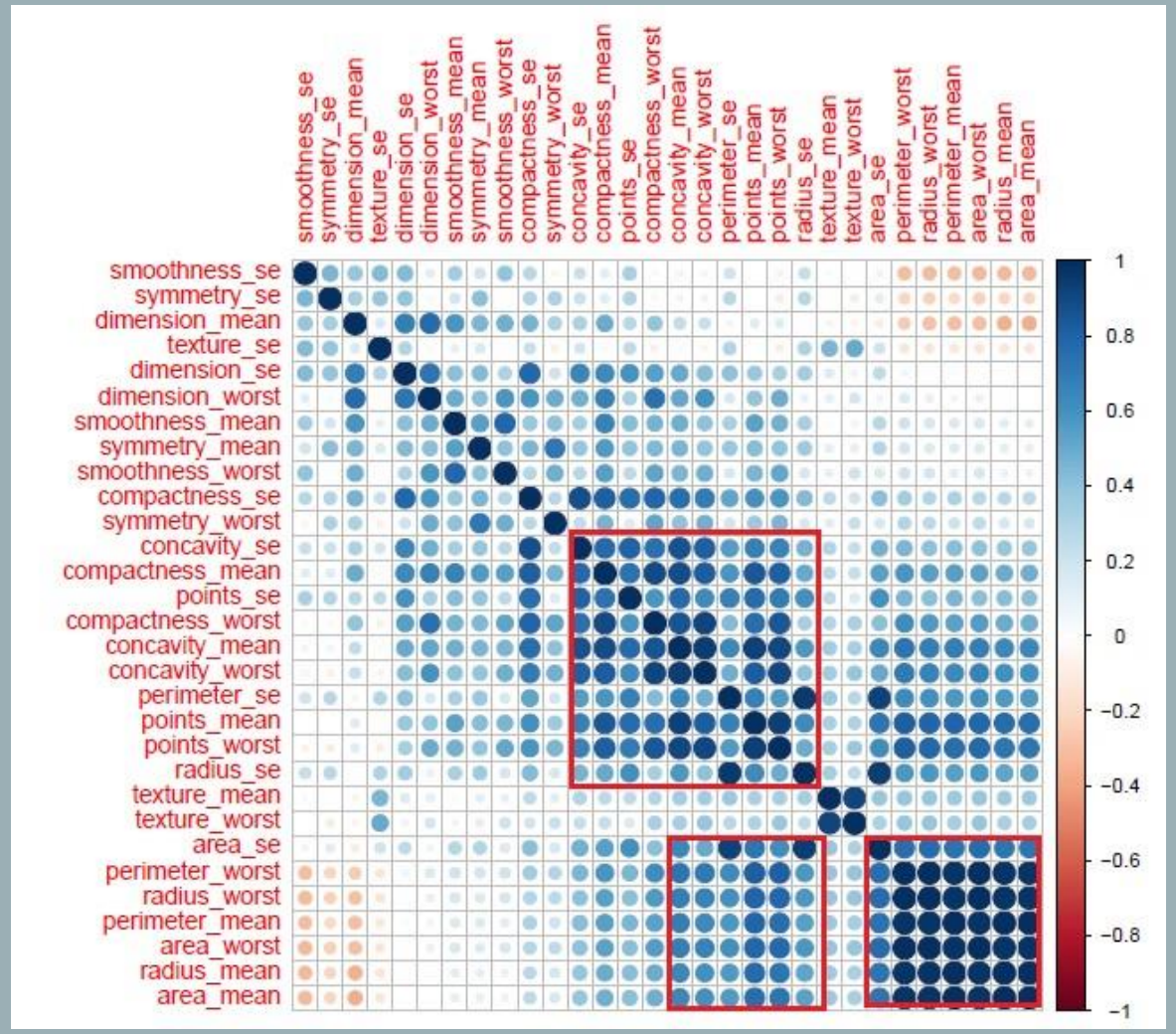


Check for NA/-Inf/Inf values

```
> apply(wbcd, 2, function(x) any(is.na(x) | is.infinite(x)))
            id      diagnosis    radius_mean   texture_mean perimeter_mean      area_mean smoothness_mean compactness_mean  concavity_mean   points_mean
         FALSE          FALSE          FALSE          FALSE          FALSE          FALSE          FALSE          FALSE          FALSE          FALSE
  symmetry_mean dimension_mean     radius_se    texture_se  perimeter_se       area_se  smoothness_se   compactness_se    concavity_se     points_se
         FALSE          FALSE          FALSE          FALSE          FALSE          FALSE          FALSE          FALSE          FALSE          FALSE
    symmetry_se   dimension_se radius_worst texture_worst perimeter_worst     area_worst smoothness_worst compactness_worst  concavity_worst   points_worst
         FALSE          FALSE          FALSE          FALSE          FALSE          FALSE          FALSE          FALSE          FALSE          FALSE
 symmetry_worst dimension_worst
         FALSE          FALSE
>
```

Exploratory Data Analysis cont.

Correlation Matrix



Data Preparation

Data normalization was done using a custom function

```
> # create normalization function
> normalize <- function(x) {
+   return ((x - min(x)) / (max(x) - min(x)))
+ }
>
>
> # normalize the wbcd data
> wbcd_n <- as.data.frame(lapply(wbcd[2:31], normalize))
> wbcd_n$diagnosis <- wbcd$diagnosis
>
> # confirm that normalization worked
> summary(wbcd_n$area_mean)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
 0.0000  0.1174  0.1729  0.2169  0.2711  1.0000
>
```

Data was split into Test and Train datasetsFirst create Benign and Malignant Dataset
Split each dataset into developing subset 80% (including training and validation subset) and testing subset 20%
Development subset had 454 obs and Test subset had 115 obs

```
> str(DeveSubset)
'data.frame':   454 obs. of  31 variables:
```

```
> str(TestingSubset)
'data.frame':   115 obs. of  31 variables:
```

Machine Learning – kNN Nearest Neighbour

- Model was trained and the applied on the test dataset
- K nearest neighbour was applied with k initially set to 20 nearest neighbours with 10 fold cross validation
- The optimum K value was found using the following formula
- Max Balanced accuracy (Mean of balanced accuracy for each of the 10 fold validation) for K neighbours
- Optimum K was 8

```
   Cell Contents
|-------------------------|
|                       N |
| Chi-square contribution |
|           N / Row Total |
|           N / Col Total |
|         N / Table Total |
|-------------------------|

=============================================
                     deve_pred
TestingSubset[, 31]          0         1   Total
---------------------------------------------
0                           72         0      72
                        15.129    26.296
                         1.000     0.000   0.626
                         0.986     0.000
                         0.626     0.000
---------------------------------------------
1                            1        42      43
                        25.332    44.030
                         0.023     0.977   0.374
                         0.014     1.000
                         0.009     0.365
---------------------------------------------
Total                       73        42     115
                         0.635     0.365
=============================================
> BA_Test
[1] 0.9883721
> table(agreement_kNN)
agreement_kNN
FALSE  TRUE
    1   114
> prop.table(table(agreement_kNN))
agreement_kNN
      FALSE        TRUE
0.008695652 0.991304348
```

Machine Learning – Support Vector Machine

Linear Kernel function

```
> CT_SVM
   Cell Contents
|-------------------------|
|                       N |
| Chi-square contribution |
|           N / Row Total |
|           N / Col Total |
|         N / Table Total |
|-------------------------|

=============================================================
                                TestingSubset$diagnosis
BreastCancer_predictions_SVM          0         1   Total
-------------------------------------------------------------
0                                    71         0      71
                                 15.855    26.548
                                  1.000     0.000   0.617
                                  0.986     0.000
                                  0.617     0.000
-------------------------------------------------------------
1                                     1        43      44
                                 25.584    42.839
                                  0.023     0.977   0.383
                                  0.014     1.000
                                  0.009     0.374
-------------------------------------------------------------
Total                                72        43     115
                                  0.626     0.374
=============================================================
> BA_Test_SVM
[1] 0.9886364
> table(agreement)
agreement
FALSE  TRUE
    1   114
> prop.table(table(agreement))
agreement
      FALSE        TRUE
0.008695652 0.991304348
>
```

Gaussian Kernel function

```
   Cell Contents
|-------------------------|
|                       N |
| Chi-square contribution |
|           N / Row Total |
|           N / Col Total |
|         N / Table Total |
|-------------------------|

=============================================================
                                TestingSubset$diagnosis
BreastCancer_predictions_rbf          0         1   Total
-------------------------------------------------------------
0                                    72         0      72
                                 16.078    26.922
                                  1.000     0.000   0.626
                                      1         0
                                  0.626     0.000
-------------------------------------------------------------
1                                     0        43      43
                                 26.922    45.078
                                  0.000     1.000   0.374
                                      0         1
                                  0.000     0.374
-------------------------------------------------------------
Total                                72        43     115
                                  0.626     0.374
=============================================================
> BA_Test_SVM_RBF
[1] 1
> agreement_rbf <- BreastCancer_predictions_rbf == TestingSubset$diagnosis
> table(agreement_rbf)
agreement_rbf
TRUE
 115
> prop.table(table(agreement_rbf))
agreement_rbf
TRUE
   1
```

Machine Learning – Naïve Bayes

```
   Cell Contents
|-------------------------|
|                       N |
| Chi-square contribution |
|           N / Row Total |
|           N / Col Total |
|         N / Table Total |
|-------------------------|


Total Observations in Table:  115


                 | TestingSubset$diagnosis
Breast_Cancer_pred |         0 |         1 | Row Total |
-------------------|-----------|-----------|-----------|
                 0 |        69 |         2 |        71 |
                   |    13.556 |    22.698 |           |
                   |     0.972 |     0.028 |     0.617 |
                   |     0.958 |     0.047 |           |
                   |     0.600 |     0.017 |           |
-------------------|-----------|-----------|-----------|
                 1 |         3 |        41 |        44 |
                   |    21.875 |    36.627 |           |
                   |     0.068 |     0.932 |     0.383 |
                   |     0.042 |     0.953 |           |
                   |     0.026 |     0.357 |           |
-------------------|-----------|-----------|-----------|
      Column Total |        72 |        43 |       115 |
                   |     0.626 |     0.374 |           |
-------------------|-----------|-----------|-----------|


> BA_Test_NB
[1] 0.9518246
> agreement <- sms_test_pred == TestingSubset$diagnosis
> table(agreement)
agreement
FALSE  TRUE
    5   110
> prop.table(table(agreement))
agreement
     FALSE       TRUE
0.04347826 0.95652174
```

Improved model - Laplace = 3

```
   Cell Contents
|-------------------------|
|                       N |
| Chi-square contribution |
|           N / Row Total |
|           N / Col Total |
|         N / Table Total |
|-------------------------|


Total Observations in Table:  115


                 | TestingSubset$diagnosis
Breast_Cancer_pred2 |         0 |         1 | Row Total |
--------------------|-----------|-----------|-----------|
                  0 |        69 |         2 |        71 |
                    |    13.556 |    22.698 |           |
                    |     0.972 |     0.028 |     0.617 |
                    |     0.958 |     0.047 |           |
                    |     0.600 |     0.017 |           |
--------------------|-----------|-----------|-----------|
                  1 |         3 |        41 |        44 |
                    |    21.875 |    36.627 |           |
                    |     0.068 |     0.932 |     0.383 |
                    |     0.042 |     0.953 |           |
                    |     0.026 |     0.357 |           |
--------------------|-----------|-----------|-----------|
       Column Total |        72 |        43 |       115 |
                    |     0.626 |     0.374 |           |
--------------------|-----------|-----------|-----------|


> BA_Test_NB2
[1] 0.9518246
> agreement <- Breast_Cancer_pred2 == TestingSubset$diagnosis
> table(agreement)
agreement
FALSE  TRUE
    5   110
> prop.table(table(agreement))
agreement
     FALSE       TRUE
0.04347826 0.95652174
```

Machine Learning – Neural Networks

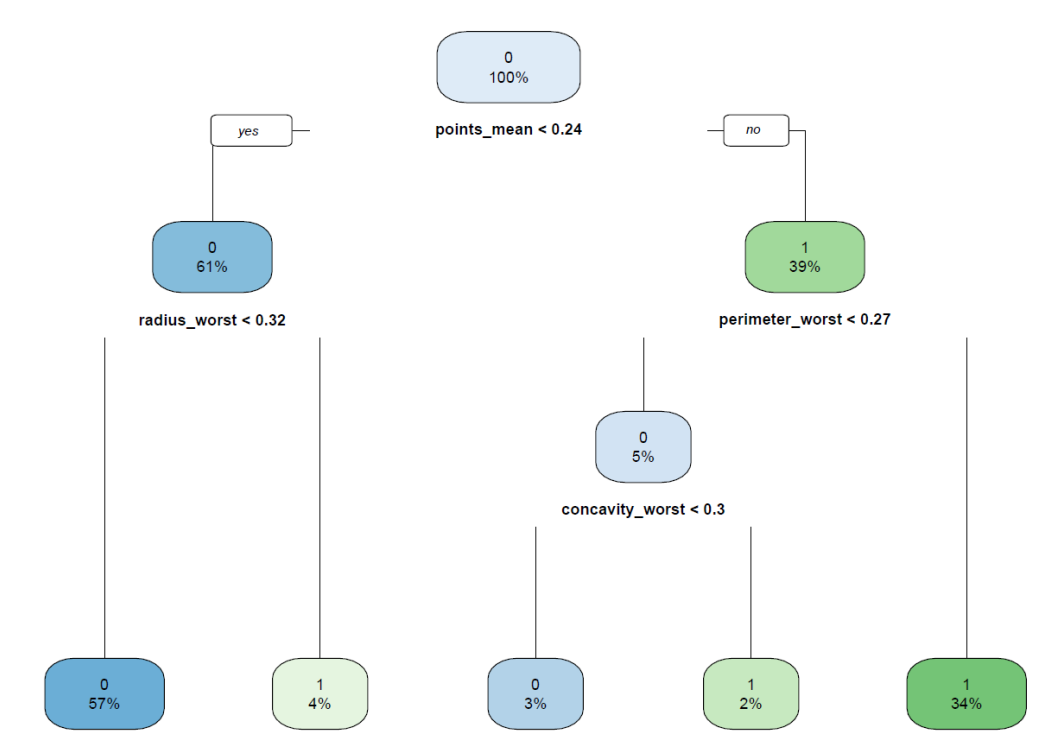
As it's a classification problem
loss set to "binary_crossentropy"
epochs set 20, batch_size = 20

```
> summary(model)
Layer (type)                 Output Shape              Param #
=================================================================
dense_3 (Dense)              (None, 256)               8192
_________________________________________________________________
dropout_2 (Dropout)          (None, 256)               0
_________________________________________________________________
dense_4 (Dense)              (None, 75)                19275
_________________________________________________________________
dropout_3 (Dropout)          (None, 75)                0
_________________________________________________________________
dense_5 (Dense)              (None, 2)                 152
=================================================================
Total params: 27,619
Trainable params: 27,619
Non-trainable params: 0
_________________________________________________________________
> |
```

```
Train on 317 samples, validate on 137 samples
Epoch 1/100
317/317 [==============================] - 2s 7ms/sample - loss: 0.5461 - acc: 0.8644 - val_loss: 0.3581 - val_acc: 0.9927
Epoch 2/100
317/317 [==============================] - 1s 3ms/sample - loss: 0.2656 - acc: 0.9890 - val_loss: 0.1472 - val_acc: 1.0000
Epoch 3/100
317/317 [==============================] - 0s 1ms/sample - loss: 0.1195 - acc: 0.9953 - val_loss: 0.0612 - val_acc: 1.0000
Epoch 4/100
317/317 [==============================] - 0s 2ms/sample - loss: 0.0542 - acc: 0.9953 - val_loss: 0.0259 - val_acc: 1.0000
Epoch 5/100
317/317 [==============================] - 1s 2ms/sample - loss: 0.0298 - acc: 1.0000 - val_loss: 0.0105 - val_acc: 1.0000
Epoch 6/100
317/317 [==============================] - 0s 1ms/sample - loss: 0.0143 - acc: 1.0000 - val_loss: 0.0046 - val_acc: 1.0000
Epoch 7/100
317/317 [==============================] - 1s 2ms/sample - loss: 0.0084 - acc: 1.0000 - val_loss: 0.0021 - val_acc: 1.0000
Epoch 8/100
317/317 [==============================] - 0s 1ms/sample - loss: 0.0040 - acc: 1.0000 - val_loss: 0.0013 - val_acc: 1.0000
Epoch 9/100
317/317 [==============================] - 1s 2ms/sample - loss: 0.0022 - acc: 1.0000 - val_loss: 6.0866e-04 - val_acc: 1.0000
Epoch 10/100
317/317 [==============================] - 0s 1ms/sample - loss: 0.0010 - acc: 1.0000 - val_loss: 2.4781e-04 - val_acc: 1.0000
Epoch 11/100
317/317 [==============================] - 1s 2ms/sample - loss: 0.0010 - acc: 1.0000 - val_loss: 1.2697e-04 - val_acc: 1.0000
Epoch 12/100
317/317 [==============================] - 0s 1ms/sample - loss: 5.6197e-04 - acc: 1.0000 - val_loss: 7.7802e-05 - val_acc: 1.0000
Epoch 13/100
317/317 [==============================] - 1s 2ms/sample - loss: 3.6274e-04 - acc: 1.0000 - val_loss: 3.3845e-05 - val_acc: 1.0000
Epoch 14/100
```

```
predictions  0  1
          0 72  0
          1  0 43
> CT_Neural_Network <- CrossTable(x = predictions, y = TestingSubset$diagnos

   Cell Contents
|-------------------------|
|                       N |
| Chi-square contribution |
|           N / Row Total |
|           N / Col Total |
|         N / Table Total |
|-------------------------|

Total Observations in Table:  115

              | TestingSubset$diagnosis
 predictions |         0 |         1 | Row Total |
-------------|-----------|-----------|-----------|
           0 |        72 |         0 |        72 |
             |    16.078 |    26.922 |           |
             |     1.000 |     0.000 |     0.626 |
             |     1.000 |     0.000 |           |
             |     0.626 |     0.000 |           |
-------------|-----------|-----------|-----------|
           1 |         0 |        43 |        43 |
             |    26.922 |    45.078 |           |
             |     0.000 |     1.000 |     0.374 |
             |     0.000 |     1.000 |           |
             |     0.000 |     0.374 |           |
-------------|-----------|-----------|-----------|
Column Total |        72 |        43 |       115 |
             |     0.626 |     0.374 |           |
-------------|-----------|-----------|-----------|


> BA_Neural_Network <- (CT_Neural_Network$t[1,1]/(CT_Neural_Network$t[1,1]+C
>
> # look only at agreement vs. non-agreement
> # construct a vector of TRUE/FALSE indicating correct/incorrect predictior
> agreement <- predictions == TestingSubset$diagnosis
> table(agreement)
agreement
TRUE
 115
> prop.table(table(agreement))
agreement
TRUE
   1
```

Machine Learning – Decision Trees




```
   Cell Contents
|-------------------------|
|                       N |
| Chi-square contribution |
|           N / Row Total |
|           N / Col Total |
|         N / Table Total |
|-------------------------|


Total Observations in Table:  115


             | TestingSubset$diagnosis
 preds_dtree |         0 |         1 | Row Total |
-------------|-----------|-----------|-----------|
           0 |        69 |         0 |        69 |
             |    15.408 |    25.800 |           |
             |     1.000 |     0.000 |     0.600 |
             |     0.958 |     0.000 |           |
             |     0.600 |     0.000 |           |
-------------|-----------|-----------|-----------|
           1 |         3 |        43 |        46 |
             |    23.113 |    38.700 |           |
             |     0.065 |     0.935 |     0.400 |
             |     0.042 |     1.000 |           |
             |     0.026 |     0.374 |           |
-------------|-----------|-----------|-----------|
Column Total |        72 |        43 |       115 |
             |     0.626 |     0.374 |           |
-------------|-----------|-----------|-----------|


> BA_Decision_Tree
[1] 0.9673913
> table(agreement)
agreement
FALSE  TRUE
    5   110
> prop.table(table(agreement))
agreement
     FALSE       TRUE
0.04347826 0.95652174
```

Machine Learning – Logistic Regression

```
> summary(cancerFitAll)

Call:
glm(formula = diagnosis ~ ., family = binomial(link = "logit"),
    data = DeveSubset)

Deviance Residuals:
       Min          1Q      Median          3Q         Max
-9.354e-04  -2.000e-08  -2.000e-08   2.000e-08   9.439e-04

Coefficients:
                  Estimate Std. Error z value Pr(>|z|)
(Intercept)        -1811.2    56802.8  -0.032    0.975
radius_mean       -66777.9  1453750.5  -0.046    0.963
texture_mean        1446.9    97339.5   0.015    0.988
perimeter_mean     23129.1  4310616.3   0.005    0.996
area_mean          48396.0  4946975.8   0.010    0.992
smoothness_mean     3451.6   180280.0   0.019    0.985
compactness_mean   -7565.5   167624.8  -0.045    0.964
concavity_mean      5309.8   206978.8   0.026    0.980
points_mean         1825.2   111750.2   0.016    0.987
symmetry_mean      -1875.6   145365.0  -0.013    0.990
dimension_mean       638.4   124135.1   0.005    0.996
radius_se          -6297.4   862986.1  -0.007    0.994
texture_se          -387.3   492037.5  -0.001    0.999
perimeter_se       -5476.1   879817.3  -0.006    0.995
area_se            34238.0  5076407.0   0.007    0.995
smoothness_se      -1596.8   375571.1  -0.004    0.997
compactness_se      6214.2   405812.7   0.015    0.988
concavity_se      -13736.7  1000726.3  -0.014    0.989
points_se           7652.7  1081525.9   0.007    0.994
symmetry_se        -3219.0   377475.9  -0.009    0.993
dimension_se       -10475.2   726295.6  -0.014    0.988
radius_worst       32460.3  2213336.6   0.015    0.988
texture_worst        715.0   382233.5   0.002    0.999
perimeter_worst     -2907.7  1537644.5  -0.002    0.998
area_worst         -31513.8  5173355.8  -0.006    0.995
smoothness_worst    -1178.1   151715.7  -0.008    0.994
compactness_worst   -4433.2   132054.1  -0.034    0.973
concavity_worst      4161.0   492298.8   0.008    0.993
points_worst         -226.3   572748.2   0.000    1.000
symmetry_worst       4813.7   260096.9   0.019    0.985
dimension_worst      4817.4   122341.9   0.039    0.969

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 5.9941e+02  on 453  degrees of freedom
Residual deviance: 1.0450e-05  on 423  degrees of freedom
AIC: 62

Number of Fisher Scoring iterations: 25
```

```
Generalized Linear Model

115 samples
 30 predictor
  2 classes: '0', '1'

No pre-processing
Resampling: Cross-Validated (10 fold, repeated 3 times)
Summary of sample sizes: 103, 104, 102, 104, 104, 104, ...
Resampling results:

  Accuracy   Kappa
  0.9514375  0.8939544

There were 50 or more warnings (use warnings() to see the first 50)
>
```

Strengths and Limitations

Model	Advantages	Disadvantages
KNN	Easy to understand and implement. Can be used both for Classification and Regression	Becomes significantly slower as the volume of data increases
SVM Linear	Performs well, If greater number of features and lesser than training sample Performs well, If a smaller number of features and large training sample Have built in multi class functionality	Memory Intensive
SVM Gaussian	Performs well If a smaller number of features and intermediate training sample Has built in multi class functionality	
Naïve Bayes	Simple supervised learning algorithms. Fast High accuracy and speed on large datasets.	Less accurate on small datasets
Neural Networks	High Accuracy Suitable for complex non-linear problems	Slow to Train
Decision Trees	Easy to interpret and explain. Simple and Fast	Can overfit Long Training Time
Logistic Regression	Many ways to regularize model Performs well, If a greater number of features than training samples Suitable for Non-Linear decision Boundaries	

Conclusion

- All the models performed very well both in terms of accuracy and execution speed due to a small test dataset.
- Support Vector Machines Gaussian Kernel and Neural Networks had the highest accuracy
- Logistical Regression had less accuracy in comparison to others
- The optimum K for k Nearest Neighbours was 8

Model	Balanced Accuracy	Agreement		Agreement Percentage	
		FALSE	TRUE	FALSE	TRUE
kNN Nearest Neighbour	0.9883	1	114	0.0869	0.9913
Support Vector Machines – Linear Kernel	0.9886	1	114	0.0869	0.9913
Support Vector Machines - Gaussian Kernel	1		115		1
Naive Bayes Classification	0.9518	5	110	0.0434	0.9565
Naive Bayes Classification - Laplace = 3	0.9518	5	110	0.0434	0.9565
Neural Networks	1		110	0	1
Decision Trees	0.9673	5	110	0.0434	0.9565
Logistic Regression	0.9514				

Further Developments

A. Add more statistical features within the dataset and perform analysis.
B. Test this model on bigger datasets and see how they behave.
C. Apply Random forest ensemble technique
D. Apply Gradient Boosting Technique

References

[1] "Breast cancer statistics | Cancer Research UK." [Online]. Available: https://www.cancerresearchuk.org/health-professional/cancer-statistics/statistics-by-cancer-type/breast-cancer. [Accessed: 28-Apr-2019].

[2] "WHO | World Cancer Report 2014," *WHO*, 2015.

[3] "Index of /ml/machine-learning-databases/breast-cancer-wisconsin." [Online]. Available: https://archive.ics.uci.edu/ml/machine-learning-databases/breast-cancer-wisconsin/. [Accessed: 28-Apr-2019].

[4] O. L. Mangasarian, W. N. Street, and W. H. Wolberg, "Breast Cancer Diagnosis and Prognosis Via Linear Programming," *Oper. Res.*, 2008.

[5] "An Introduction to corrplot Package." [Online]. Available: https://cran.r-project.org/web/packages/corrplot/vignettes/corrplot-intro.html. [Accessed: 28-Apr-2019].

[6] T. M. Cover and P. E. Hart, "Nearest neighbor pattern classification. IEEE Trans Inf Theory IT-13(1):21-27," *IEEE Trans. Inf. Theory*, 1967.

[7] S. D. Jadhav and H. Channe, "Comparative Study of K-NN , Naive Bayes and Decision Tree Classification Techniques." 2016.

[8] "Chapter 2 : SVM (Support Vector Machine) — Theory – Machine Learning 101 – Medium." [Online]. Available: https://medium.com/machine-learning-101/chapter-2-svm-support-vector-machine-theory-f0812effc72. [Accessed: 28-Apr-2019].

[9] "Lecture7 SVM =nc." .

References

[10] “Naive Bayes Classification using Scikit-learn (article) - DataCamp.” [Online]. Available: https://www.datacamp.com/community/tutorials/naive-bayes-scikit-learn. [Accessed: 15-Apr-2019].
[11] N. Friedman, D. Geiger, and M. Goldszmit, “Bayesian Network ClassifiersOverfitting and Underfitting With Machine Learning Algorithms (no date). Available at: https://machinelearningmastery.com/overfitting-and-underfitting-with-machine-learning-algorithms/ (Accessed: 1 July 2018).,” *Mach. Learn.*, 1997.
[12] “Chapter 1. What is deep learning? - Deep Learning with R.” [Online]. Available: https://livebook.manning.com/#!/book/deep-learning-with-r/chapter-1/38. [Accessed: 28-Apr-2019].
[13] “Decision Trees in Machine Learning – Towards Data Science.” [Online]. Available: https://towardsdatascience.com/decision-trees-in-machine-learning-641b9c4e8052. [Accessed: 28-Apr-2019].
[14] “How to Perform a Logistic Regression in R | DataScience+.” [Online]. Available: https://datascienceplus.com/perform-logistic-regression-in-r/. [Accessed: 28-Apr-2019].
[15] “Logit Regression | R Data Analysis Examples.” [Online]. Available: https://stats.idre.ucla.edu/r/dae/logit-regression/. [Accessed: 28-Apr-2019].
[16] “R: Fitting Generalized Linear Models.” [Online]. Available: https://stat.ethz.ch/R-manual/R-patched/library/stats/html/glm.html. [Accessed: 28-Apr-2019]