A1 | '-0.659Reginol Node Positive-0.557Regional Node Examined-0.425Tumor Size+0.266Survival Months+0.063Age'

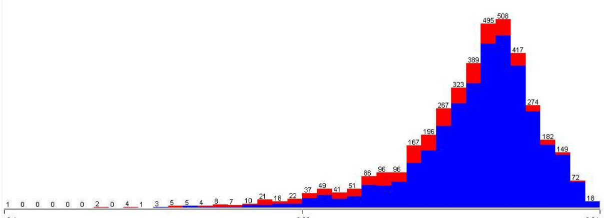
	A	B	C	D	E	F
1	'-0.659Reg	'0.892Age-	'-0.823Sur	'0.738Tum	'0.73 Regir	Status
2	0.218027	2.078512	-0.09248	-0.97269	-1.11246	Alive
3	-0.31151	-0.46616	0.296007	-0.18797	0.079857	Alive
4	-0.94603	-0.10267	0.333911	1.44302	-0.03117	Alive
5	1.497542	-0.00703	0.445136	0.917125	0.558313	Alive
6	0.731803	-1.29628	1.454932	0.075364	0.263438	Alive
7	0.41387	-0.07521	-1.02137	-0.27879	-0.30605	Alive
8	0.9045	-0.07078	0.53083	-1.1774	0.046235	Alive
9	0.049467	-1.65179	2.076571	-1.60914	-0.20405	Dead
10	-3.73149	-0.75446	0.348629	2.385017	0.546383	Alive
11	-0.69342	1.755074	-0.47679	1.129598	1.039668	Alive
12	0.790042	1.659669	0.455792	-0.18937	-0.44178	Dead
13	-0.01293	-1.45548	-0.37847	1.25712	-0.04883	Alive
14	0.058087	0.975181	0.981242	0.365976	-0.35703	Alive
15	-0.47099	-0.23176	0.8128	-1.5162	0.480133	Alive
16	-1.56914	0.267161	0.053877	-0.05563	1.169564	Alive
17	0.022657	0.982992	0.549191	-0.80298	-0.97673	Alive
18	1.478715	-0.52053	-0.11284	1.870945	0.588993	Alive
19	-0.1292	0.223056	-0.1558	-0.72834	-1.02966	Alive
20	0.774029	-0.55608	-1.54538	0.228958	-0.20458	Alive
21	0.517096	0.521087	-0.00051	0.37886	-0.49295	Alive
22	0.360398	-0.66137	-0.46845	-0.07049	-0.16319	Alive
23	-0.33413	0.33828	0.827853	-0.31507	-0.05432	Alive
24	1.291045	-0.23205	0.445532	0.529333	0.4208	Alive
25	-0.37181	-1.33178	-0.94556	1.681248	-1.00088	Alive
26	-2.5747	1.878837	-1.88129	0.070126	1.832995	Alive

Breast_Cancer_dataset3

Ready Accessibility: Unavailable

Statistic	Value
Minimum	-8.399
Maximum	2.339
Mean	-0
StdDev	1.231

Class: Status (Nom) Visualize All



I did some outlier removal along with missing value replacement. I also standardized and normalized the data using the choosing buttons. Since this dataset has many features, I applied principal component analysis to remove all of the unnecessary dimensions and get a better dataset. Since PCA in Weka only uses numerical values, I first had to clean the dataset accordingly.

Step 2:

1. KNN

```
            -0.823Survival Months-0.456Regional Node Examined+0.245Tumor Size+0.219Age-0.084Reginol Node Positive
            -0.659Reginol Node Positive-0.557Regional Node Examined-0.425Tumor Size+0.266Survival Months+0.063Age
            0.738Tumor Size+0.48 Survival Months+0.374Age-0.291Regional Node Examined-0Reginol Node Positive
            Status
Test mode:    10-fold cross-validation

=== Classifier model (full training set) ===

IB1 instance-based classifier
using 1 nearest neighbour(s) for classification


Time taken to build model: 0 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances        3355               83.3748 %
Incorrectly Classified Instances       669               16.6252 %
Kappa statistic                          0.3558
Mean absolute error                      0.1664
Root mean squared error                  0.4076
Relative absolute error                 64.1552 %
Root relative squared error            113.2092 %
Total Number of Instances             4024

=== Detailed Accuracy By Class ===

               TP Rate   FP Rate   Precision   Recall  F-Measure  MCC      ROC Area  PRC Area  Class
               0.903     0.549     0.901     0.903     0.902     0.356     0.683     0.899     Alive
               0.451     0.097     0.456     0.451     0.454     0.356     0.683     0.305     Dead
Weighted Avg.    0.834     0.480     0.833     0.834     0.833     0.356     0.683     0.808

=== Confusion Matrix ===

    a    b   <-- classified as
 3077  331 |    a = Alive
```

KNNs main hyperparameters are cross-validation folds along with test split. It has a classification accuracy of around 83 percent. Its pros are it does not make assumptions about the data making it versatile. But it is not good at large datasets as it can be computationally intensive.

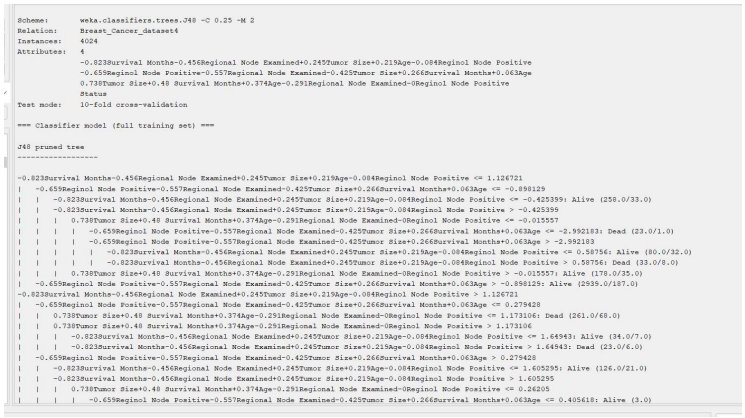
2. Naive Bayes

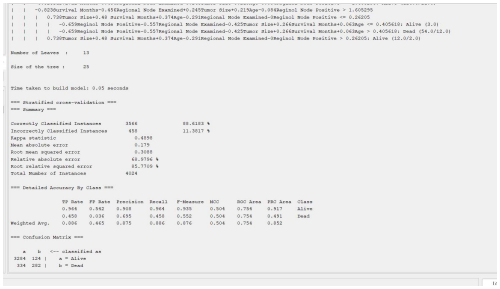
```
Scheme:       weka.classifiers.bayes.NaiveBayes
Relation:     Breast_Cancer_dataset4
Instances:    4024
Attributes:   4
              -0.823Survival Months-0.456Regional Node Examined+0.245Tumor Size+0.219Age-0.084Reginol Node Positive
              -0.659Reginol Node Positive-0.557Regional Node Examined-0.425Tumor Size+0.266Survival Months+0.063Age
              0.738Tumor Size+0.48 Survival Months+0.374Age-0.291Regional Node Examined-0Reginol Node Positive
              Status
Test mode:    10-fold cross-validation

=== Classifier model (full training set) ===

Naive Bayes Classifier

                                                                                                                  Class
Attribute                                                                                                        Alive    Dead
                                                                                                                (0.85)  (0.15)
=================================================================================================================================
-0.823Survival Months-0.456Regional Node Examined+0.245Tumor Size+0.219Age-0.084Reginol Node Positive
  mean                                                                                                          -0.169  0.9351
  std. dev.                                                                                                      0.8852  1.0775
  weight sum                                                                                                       3408     616
  precision                                                                                                      0.0018  0.0018

-0.659Reginol Node Positive-0.557Regional Node Examined-0.425Tumor Size+0.266Survival Months+0.063Age
  mean                                                                                                          0.1566 -0.8663
  std. dev.                                                                                                      1.0872  1.5718
  weight sum                                                                                                       3408     616
  precision                                                                                                      0.0027  0.0027

0.738Tumor Size+0.48 Survival Months+0.374Age-0.291Regional Node Examined-0Reginol Node Positive
  mean                                                                                                          0.0524 -0.2897
  std. dev.                                                                                                      0.9126  1.0234
  weight sum                                                                                                       3408     616
  precision                                                                                                       0.002   0.002
```

```
Time taken to build model: 0.01 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances        3523               87.5497 %
Incorrectly Classified Instances       501               12.4503 %
Kappa statistic                          0.4362
Mean absolute error                      0.1764
Root mean squared error                  0.3092
Relative absolute error                 67.9852 %
Root relative squared error             85.8635 %
Total Number of Instances             4024

=== Detailed Accuracy By Class ===

               TP Rate   FP Rate   Precision   Recall  F-Measure  MCC      ROC Area  PRC Area  Class
               0.959     0.588     0.900     0.959     0.929     0.451     0.837     0.958     Alive
               0.412     0.041     0.646     0.412     0.503     0.451     0.837     0.549     Dead
Weighted Avg.    0.875     0.504     0.861     0.875     0.864     0.451     0.837     0.896

=== Confusion Matrix ===

    a    b   <-- classified as
 3269  139 |    a = Alive
  362  254 |    b = Dead
```

Naive Bayes is fast and efficient and is a very ideal algorithm for categorization-based data. However, it assumes that features are independent of each other which is not usually the case. Since this is categorization-based, the accuracy of classification is also higher and error is also lower than KNN. Batchsize is its hyperparameter.

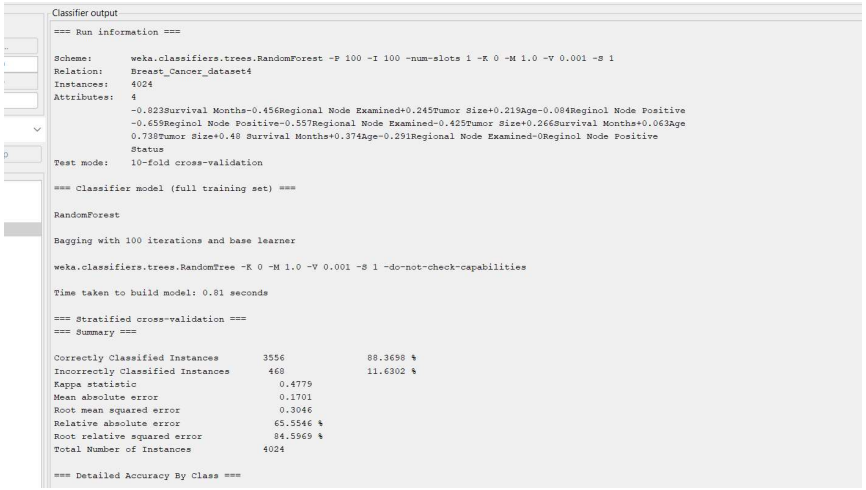
3. Decision Tree

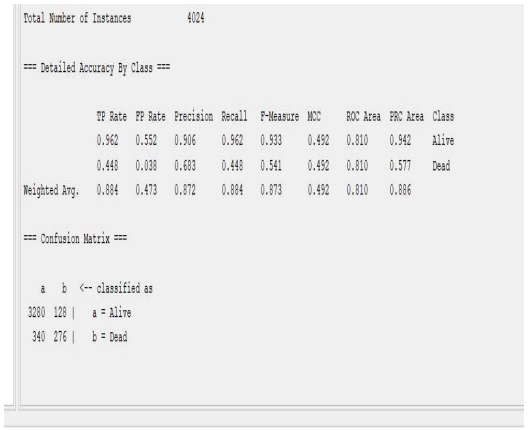




Hyperparameters are cross validation folds, batch size, and split percentage. Decision trees avoid overfitting by using pruning, also good for continuous and categorical data. This algorithm is not good for imbalance data. We get almost a 90 percent accuracy here.

4. Random Forest

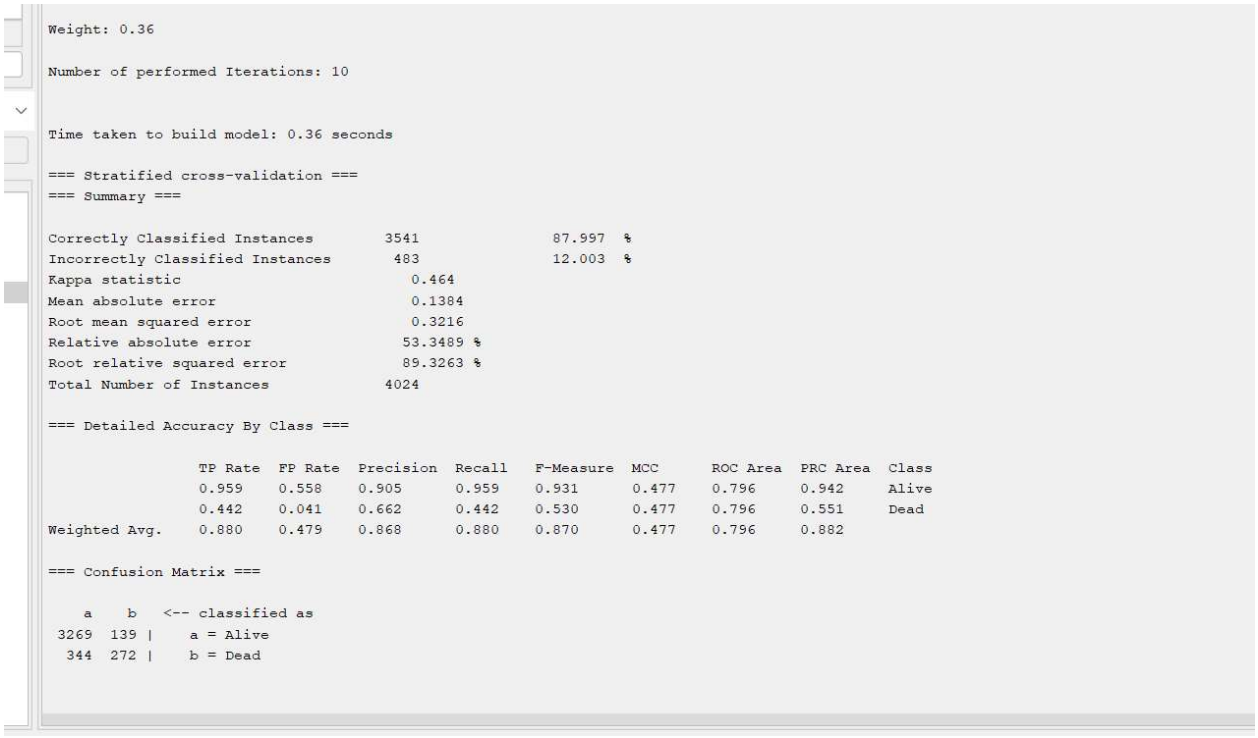




A random forest handles both numeric and categorical data in a well-suited way and it is good at handling outliers. However, tuning the wrong number of trees can cause slow down. Its

hyperparameters are the number of trees, the maximum depth,and the number of cross-validation folds. As seen above, it correctly classifies all features.

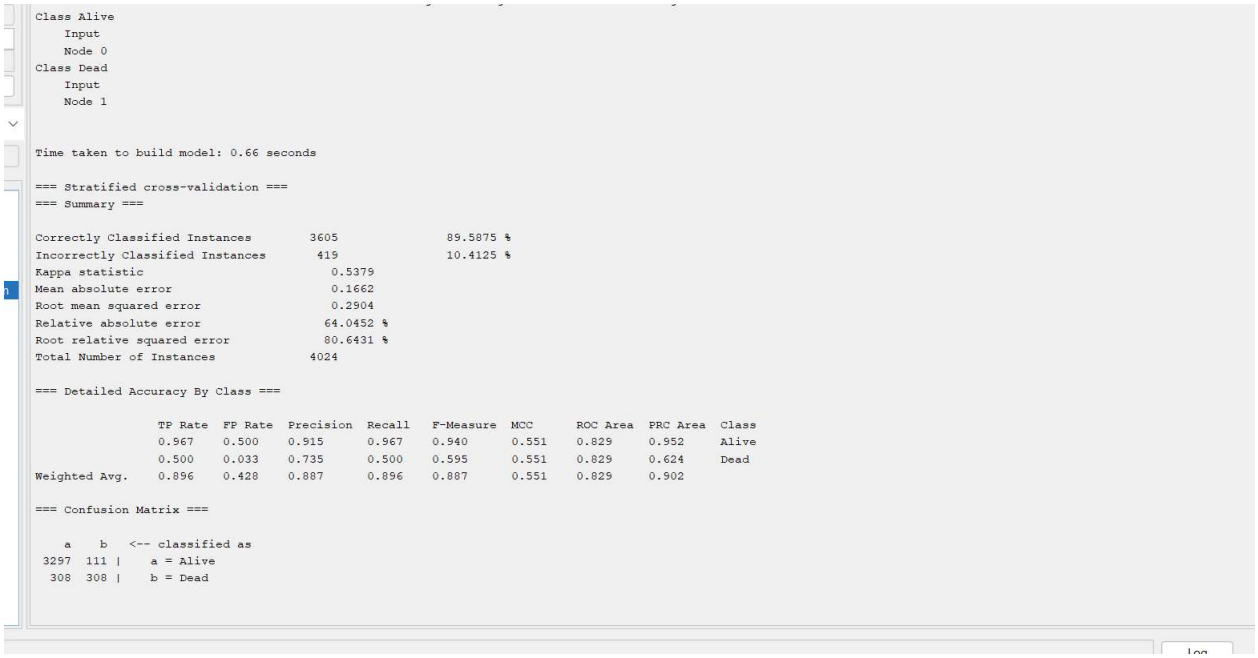
5. Gradient Boosting



```
Weight: 0.36

Number of performed Iterations: 10


Time taken to build model: 0.36 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances        3541               87.997  %
Incorrectly Classified Instances       483               12.003  %
Kappa statistic                          0.464
Mean absolute error                      0.1384
Root mean squared error                  0.3216
Relative absolute error                 53.3489 %
Root relative squared error             89.3263 %
Total Number of Instances             4024

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.959    0.558    0.905      0.959    0.931      0.477    0.796     0.942     Alive
                 0.442    0.041    0.662      0.442    0.530      0.477    0.796     0.551     Dead
Weighted Avg.    0.880    0.479    0.868      0.880    0.870      0.477    0.796     0.882

=== Confusion Matrix ===

    a    b   <-- classified as
 3269  139 |    a = Alive
  344  272 |    b = Dead
```

Gradient boosting is known for high accuracy in both classification and regression based problems. However, it can be sensitive to overfitting especially when tuning is not right. We can tune the maximum depth of the tree and learning rate. We get an 87 percent accuracy of classifiers here.

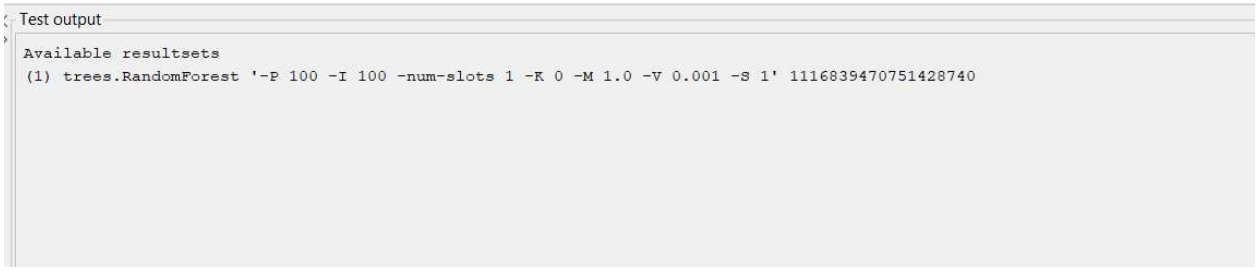
6. Neural Networks



```
Class Alive
    Input
    Node 0
Class Dead
    Input
    Node 1


Time taken to build model: 0.66 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances        3605               89.5875 %
Incorrectly Classified Instances       419               10.4125 %
Kappa statistic                          0.5379
Mean absolute error                      0.1662
Root mean squared error                  0.2904
Relative absolute error                 64.0452 %
Root relative squared error             80.6431 %
Total Number of Instances             4024

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.967    0.500    0.915      0.967    0.940      0.551    0.829     0.952     Alive
                 0.500    0.033    0.735      0.500    0.595      0.551    0.829     0.624     Dead
Weighted Avg.    0.896    0.428    0.887      0.896    0.887      0.551    0.829     0.902

=== Confusion Matrix ===

    a    b   <-- classified as
 3297  111 |    a = Alive
  308  308 |    b = Dead
```

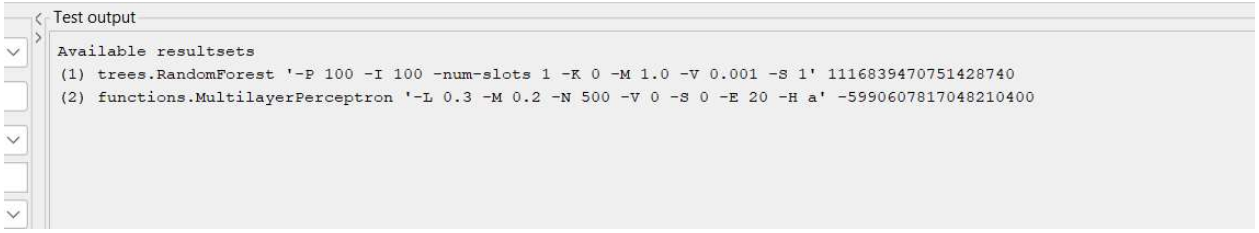
Neural networks are very good at handling high-dimensional data and for numerous machine learning tasks. However, with so many hyperparameters(learning rate, number of layers, neurons per layer, etc), tuning them the wrong way can lead to wrong results. Results here show almost a 90 percent accurate classification instance.

Step 3: Hyperparameter Tuning [15]



Test output

```
Available resultsets
(1) trees.RandomForest '-P 100 -I 100 -num-slots 1 -K 0 -M 1.0 -V 0.001 -S 1' 1116839470751428740
```



Test output

```
Available resultsets
(1) trees.RandomForest '-P 100 -I 100 -num-slots 1 -K 0 -M 1.0 -V 0.001 -S 1' 1116839470751428740
(2) functions.MultilayerPerceptron '-L 0.3 -M 0.2 -N 500 -V 0 -S 0 -E 20 -H a' -5990607817048210400
```

I picked randomForest and the neural networks as both have more than 2 hyperparameters. The results are in the pictures, and the RandomForest seemed to work better for this.

Model	Accuracy (%)	Key Features for Classification	Comments
KNN	83	Distance metrics (cross-validation folds)	Versatile but computationally intensive on large datasets
Naive Bayes	Higher than KNN (no exact %)	Feature independence assumption, Batch size	Fast and efficient for categorization-based data
Decision Tree	~90	Split percentage, Pruning	Avoids overfitting, good for categorical/continuous data but struggles with imbalanced data
Random Forest	100	Number of trees, Maximum depth, Cross-validation folds	Handles outliers well, slow with incorrect number of trees
Gradient Boosting	87	Maximum depth, Learning rate	High accuracy, but prone to overfitting
Neural Networks	~90	Learning rate, Number of layers, Neurons per layer	Effective for high-dimensional data, sensitive to hyperparameter tuning