Prediction of Breast Cancer Survival

Practical Data Science (COSC1295)

22nd May 2018

Team Members:

Janarth Punniyamoorthy

(s3706154@student.rmit.edu.au)

Charles Galea

(s3688570@student.rmit.edu.au)

Lecturer:

Dr. Yongli Ren

Table of Contents

Abstract

Breast cancer is a leading cause of cancer deaths in women aged 20 to 59 years. Early diagnosis can result in significantly improved chances of long term survival and has resulted in a dramatic reduction in worldwide breast cancer mortality in recent years. Besides early detection, it is also important to identify high risk individuals having poor chances of long term survival, so that they can be targeted for immediate therapy. Considering this, we hypothesize that a machine learning classification model trained on patient clinical records could be used to quickly identify high risk individuals. In this study we have tested three prediction classification models (SVM, *k*NN and Decision Trees) for the identification of high risk cancer patients.

Introduction

Breast cancer is a common cancer accounting for 1 in every 4 cancer diagnoses amongst women. It is estimated that in 2018 over 18,000 Australians will be diagnosed with breast cancer and more than 3,000 will die from this disease (https://breast-cancer.canceraustralia.gov.au/statistics). Early diagnosis can lead to a dramatic increase in long term survival rates from 56% to more than 86% (Harbeck and Gnant, 2017, Siegel et al., 2016). It has been reported that if the primary cancer has not spread beyond the breast to the lymph nodes there is a significantly better prognosis. Metastatic breast cancers which have spread to the lymph nodes and more distant sites account for 90% of all deaths from this disease (Fouad et al., 2015, Peart, 2017). In this report, we have explored whether historical patient medical records, including a patient's age and the number of auxiliary lymph nodes containing cancer cells (an indication that the cancer has metastasized), could be used to predict whether a patient is at high risk of dying within 5 years. From a clinical perspective, it would be beneficial to identify these patients so that they can be targeted for early therapeutic treatment.

Methodology

Data

The Haberman's Survival dataset (Haberman, 1976) was obtained from the University of California, Irvine Machine Learning Repository (https://archive.ics.uci.edu/ml/datasets/l). The dataset contained clinical data from a study conducted between 1958 and 1970 at the University of Chicago's Billings Hospital for 306 patients who had undergone surgery for breast cancer. The data comprised: patient age at time of operation, year of operation, the number of auxiliary lymph nodes detected with cancer and the patient survival status (denoted as 1 for patients surviving beyond 5 years or 2 for patients who did not) (**Table 1**).

Data Analysis

Data manipulation and statistical analyses were performed using the python packages pandas (https://pandas.pydata.org/) and NumPy (http://www.numpy.org/). The data were initially checked and corrected for errors and missing data points. Statistical parameters (mean, standard deviation, minimum, maximum, quantiles) were then determined for the entire patient cohort as well as for each target group (based on survival status).

Data Visualization

The python plotting package Matplotlib (https://matplotlib.org/) was used for data visualization.

Machine Learning

Support Vector Machine (SVM)

SVM is a discriminant model which tries to find the combination of samples to build a plain maximizing the margin between the two classes. It is a supervised learning technique for classification and regression that can automatically fit linear and non-linear models (Boschetti and Massaron, 2015). SVM is available as part of the python *Sklearn* python library and can be used for classification (SVC) and regression (SVR). When classes are not separable in feature space by a linear plain the SVM classifier can use a non-linear (kernel) decision function (e.g. polynomial). Hyperparameters that need optimization included: kernel, gamma (kernel coefficient) and C (penalty parameter for the error term). The SVM model is defined as :

$$M_{a,w0}(q) = \sum_{i=1}^{s}(t_i \times a[i] \times (d_i \cdot q) + w_0)$$

where q is the set of descriptive features for a query instance; $(d_1, t_1), \ldots \ldots .., (d_{s,}, t_s)$ are s support vectors (instances composed of descriptive features and a target feature); w_0 is the first weight of the decision boundary; and a is a set of parameters determined during the training process (there is a parameter for each support vector $a[1], \ldots \ldots .., a[s]$) (Kelleher et al., 2015). When the output of this equation if greater than 1 we predict the positive target level for the query, and when the output is less than -1 we predict the negative target level.

k-Nearest Neighbours Classifier

The *k*NN classifier defines the class of a test instance based on the majority vote for its *k*-nearest neighbours derived from the training data (Friedman et al., 2001). Due to the relatively small size of the dataset (4 features with 306 rows), *k*NN hyperparameters (*k* – n-neighbours, distance metric p and weights) were optimized using 5-fold cross validation, to achieve the best balance between the bias and variance for the model. The distance metric was determined by the following equation:

$$\left(\sum_{i=1}^{n}|x_i - y_i|^p\right)^{1/p}$$

where $X = (x_1, x_2, \ldots \ldots .., x_n)$ and $Y = (y_1, y_2, \ldots \ldots .., y_n) \in \mathbb{R}^n$.

For 5-fold cross-validation the data was divided into 5 folds and the *k*NN model was applied to make predictions on the 5th segment. Predictions were performed using the equation:

$$y = \frac{1}{K}\sum_{l=1}^{k} y_i$$

Where y_i is the ith case of the sample and y is the prediction of the query point.

The process was then successively applied to the other 4 segments. The computed errors were then averaged to yield a measure of performance for the model. These steps were repeated for each parameter and the value achieving the lowest error (or highest classification accuracy) was chosen as the optimal value. The *k*NN classifier (*K*NeighborsClassifier) was implemented using the python sklearn package (Pedregosa et al., 2011).

Data scaling for the SVM and *k*NN classifiers

The SVM and *k*NN classifiers were tested on the dataset following feature scaling to compensate for differences in the measurement scale used of the various features. Two scaling methods (standardization and normalization) (Kelleher et al., 2015) were used to rescale each feature prior to applying the SVM and *k*NN classifiers.

The data were standardized to a standard normal distribution with $\mu = 0$ and $\sigma = 1$ using the formula:

$$z = \frac{x - \mu}{\sigma}$$

where z is the standard score (also called the z-score), μ is the mean (average) and σ is the standard deviation from the mean. The mean was calculated using the equation:

$$\mu = \frac{1}{N}\sum_{i=1}^{N}(x_i)$$

where N is the total number of observations and x_i is the ith observation. The standard deviation was derived from the equation:

$$\sigma = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(x_i - \mu)^2}$$

The data were normalized (also referred to as Min-Max scaling) by applying the following formula:

$$X_{norm} = \frac{X - X_{min}}{X_{max} - X_{min}}$$

where X_{norm} is the normalized feature value, X is the original value while X_{min} and X_{max} are the minimal and maximum values of the feature, respectively.

Decision Trees Classifier

The decision tree was built by partitioning instances into local subsets by performing binary splits recursively using the most significant variable and value of the variable which gives the best homogeneous sets. By default, we used the *Gini method* to determine the homogeneity of sets. The Gini method compares the homogeneity of sets by computing the sum of squares of probability for success and failure.

$$(p^2 + q^2)$$

Parameter constraints were applied to the decision tree model to prevent overfitting the dataset. The model hyperparameters depth, minimum required samples in a node to split, minimum required samples to be in a leaf node, and maximum number of leaf nodes were optimized to obtain the highest prediction performance while avoiding overfitting.

Hyperparameter Optimization using Grid Search

Model hyperparameters were optimized by a grid search approach using the *GridSearchCV* algorithm that is available in the python *Sklearn* library. *GridSearchCV* uses stratified cross-validation which ensures that each fold has the same proportion of observations with a given target value.

Model Evaluation

Model performance was evaluated using the following parameters:

$$classification\ error\ rate = \frac{(FP + FN)}{(TP + TN + FP + FN)}$$

$$classification\ accuracy = \frac{(TP + TN)}{(TP + TN + FP + FN)}$$

$$precision = \frac{TP}{(TP + FP)}$$

$$recall = \frac{TP}{(TP + FN)}$$

$$F1\ measure = 2 \times \frac{(precision \times recall)}{(precision + recall)}$$

where FP was the number of false positives (negative target feature instances incorrectly predicted as positive), FN was the number of false negatives (positive target feature instances incorrectly predicted as negative), TP was the number of true positives (correctly predicted positive target feature instances) and TN was the number of true negatives (correctly predicted negative target feature instances).

The average class accuracy was used to determine classification accuracy to account for the imbalance between patient survival status values in the target feature.

$$average\ class\ accuracy = \frac{1}{\frac{1}{|levels(t)|}\sum_{l \in levels(t)} \frac{1}{recall_l}}$$

where $levels(t)$ was the set of levels that the target feature, t, could assume; $|levels(t)|$ was the size of the set; and $recall_l$ refers to the recall achieved by the model for level l (Kelleher et al., 2015).

Results

Data Pre-processing

Statistical Analyses

Data collected for 306 breast cancer patients (Haberman, 1976) was used to train several machine learning classification algorithms for the prediction of cancer patient survival. The data consisted of the features: patient age, year of operation, number of positive auxiliary lymph nodes containing cancer cells and survival status (**Table 1**). Approximately two-thirds (73.5%) of the breast cancer patient cohort survived for 5 years or longer while a third (26.5%) had succumbed to the disease with 5 years.

Table 1. Haberman Breast Cancer Dataset

Number of patients	306
Patient ages	30-83
Year of operation	1958-1969
Number of nodes	0-52
Patient survival status	
Less than 5 years	81 (73.5%)
5 years or longer	225 (26.5%)

We initially examined the data distribution for each feature for the patient cohort. Patient ages were normally distributed with a mean value of 52.46 years and standard deviation of 10.80 (**Table 2 and**

Fig. 1A). The youngest patient was 30 years old while the eldest was 83. The data indicated that there was little correlation between a patient's age and their long-term survival (**Fig. 2B**).

Table 2. Statistical parameters for the entire breast cancer patient cohort.

	Patient Age	Year of Operation	Number of Nodes	Survival Status
count	306	306	306	306
mean	52.46	62.85	4.03	1.26
std	10.80	3.25	7.19	0.44
min	30.0	58.0	0.0	1.0
25%	44.0	60.0	0.0	1.0
50%	52.0	63.0	1.0	1.0
75%	60.8	65.8	4.0	2.0
max	83.0	69.0	52.0	2.0

Similar numbers of patients (between 25 and 30) aged between 45 and 55 years were operated on between 1958 and 1969 (**Figs. 1B and 2A**). Interesting, in 1959 nearly twice the number of patients underwent an operation with a significant number of these being younger cancer patients (~ 40 years old) (**Fig. 2A**).

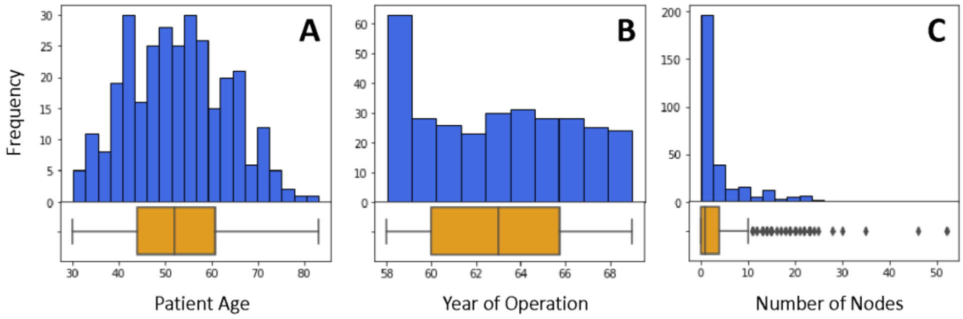


Fig. 1. **(A)** Patient ages, **(B)** year of operation and **(C)** number of auxiliary lymph nodes containing cancer cells for patients in the breast cancer cohort. The data were plotted as a histogram **(top)** and boxplot **(bottom)** to highlight the distribution of the data.

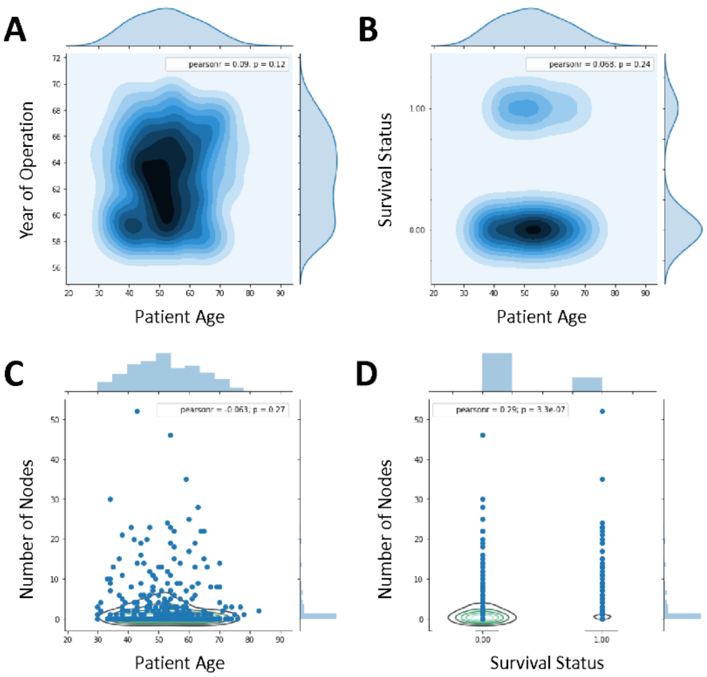


Fig. 2. 2D contour plots examining the distribution of breast cancer patient ages, year of operation and number of auxiliary nodes containing cancer cells.

No auxiliary nodes containing cancer cells were detected in the majority of breast cancer patients (**Figs. 1C & 2C**). The remaining patients had less than 25 positive nodes while several patients had 25 or more positive nodes. Interestingly, survival status did not appear to correlate with the number of positive auxiliary nodes (**Fig. 2D**). The range of positive auxiliary nodes detected was similar for those surviving for 5 years or longer and individuals surviving for less than 5 years (**Fig. 2D**).

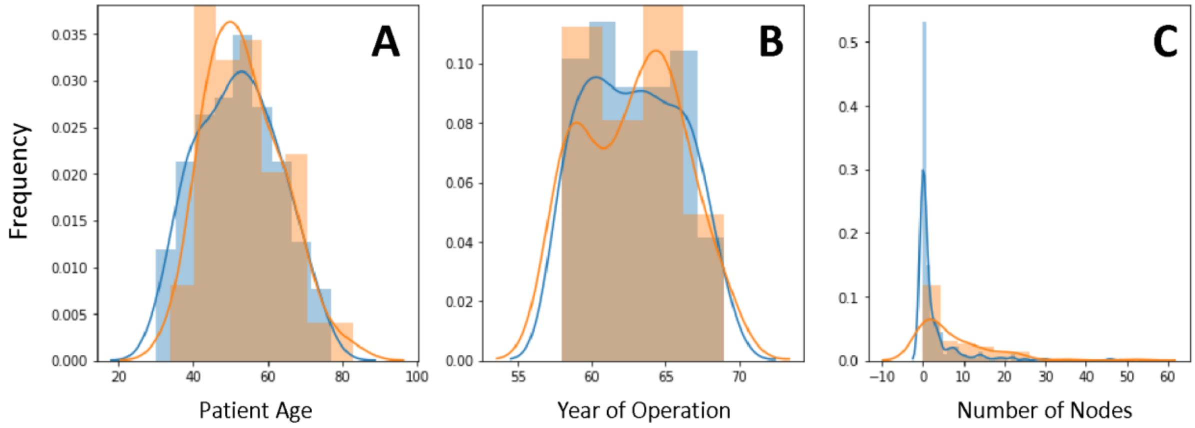


Fig. 4. Comparison of the distribution of **(A)** patient ages, **(B)** year of operation and **(C)** number of positive auxiliary nodes for breast cancer patients surviving for 5 years or longer (blue) or less than 5 years (orange). The data were plotted as a histogram overlaid with the corresponding density plot (solid lines).

The distribution of patient ages for those undergoing surgery each year, based on survival status, were similar (**Fig. 4A & B and 5A & B**) with comparable mean and standard deviation values (**Tables 2 to 4**). Patients surviving for less than 5 years appeared to have more nodes compared to those surviving for longer (**Fig. 4C & 5C and Tables 3 & 4**).

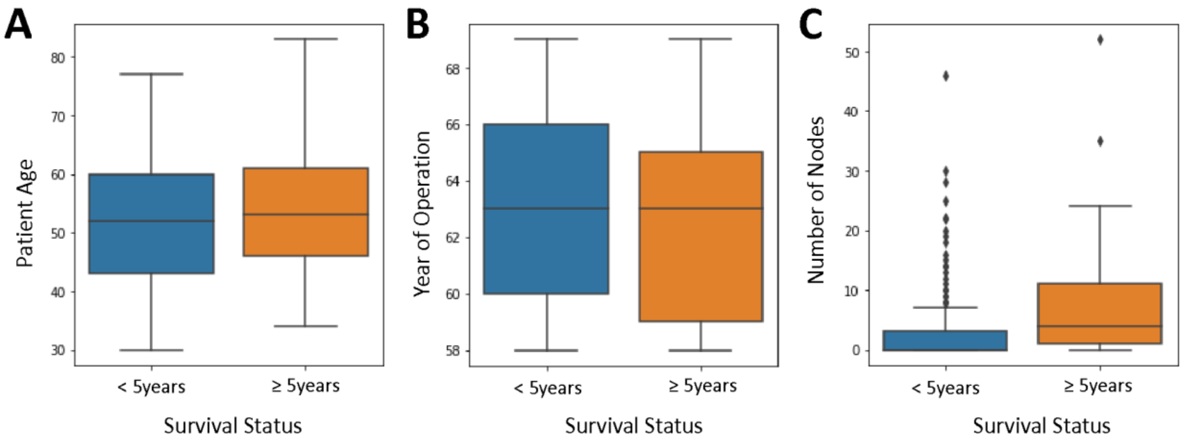


Fig. 5. Comparison of survival status (blue - < 5 years and orange - > 5 years) based on **(A)** patient age, **(B)** year of operation and **(C)** number of positive auxiliary nodes.

Table 3. Statistical parameters for breast cancer patients surviving for 5 years or longer.

	Patient Age	Year of Operation	Number of Nodes	Survival Status
count	225	225	225	225
mean	52.02	62.86	2.79	1.00
std	11.01	3.22	5.87	0.00
min	30.0	58.0	0.0	1.0
25%	43.0	60.0	0.0	1.0
50%	52.0	63.0	0.0	1.0
75%	60.0	66.0	3.0	1.0
max	77.000000	69.000000	46.000000	1.0

Table 4. Statistical parameters for the breast cancer patients surviving for less than 5 years.

	Patient Age	Year of Operation	Number of Nodes	Survival Status
count	81	81	81	81
mean	53.68	62.83	7.46	2.00
std	10.17	3.34	9.18	0.00
min	34.0	58.0	0.0	2.0

	Patient Age	Year of Operation	Number of Nodes	Survival Status
25%	46.0	59.0	1.0	2.0
50%	53.0	63.0	4.0	2.0
75%	61.0	65.0	11.0	2.0
max	83.0	69.0	52.0	2.0

Data pre-processing

Feature transformation

The histogram plotting the distribution of positive auxiliary nodes was highly right skewed due to the presence of several outliers (**Figs. 1C & 4C**). Since the *k*NN classifier is a 'distance-based' predictor the presence of outliers can have a detrimental effect upon its performance. To minimize the effect of these outliers, the data were transformed with several different transformation functions (i.e. $\log_{10}$, $\log_2$, square root, cubed root) to attempt to obtain a more normal distribution. The best results were obtained by applying a cubed root function, resulting in an overall improvement in the distribution of the plotted data (**Fig. 6**).

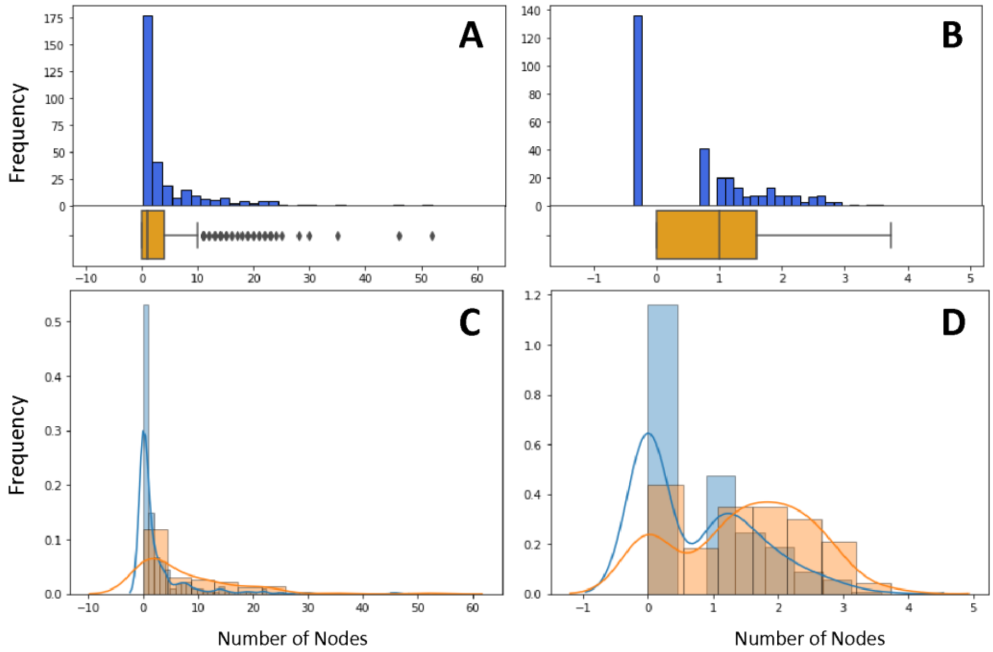


Fig. 6. Transformation of data for the 'number of positive auxiliary nodes' feature resulted in an improved distribution. The highly right skewed data for the number of positive auxiliary nodes **(A)** was processed with a cubed root function **(B)**. The histogram is show at top while the corresponding boxplot is at the bottom of the figure. **(C)** and **(D)** show the histograms and density plots (solids lines) for breast cancer patients surviving for 5 years or longer (blue) or less than 5 years (orange).

Feature Scaling

We standardization (or normalization) the dataset to compensate for differences in the range of measurements used for each feature (**Figs. 7**). This ensured the data were compared over similar ranges while having negligible effect upon the overall distribution of the data (**Figs. 8 and 9**).

Feature Scaling

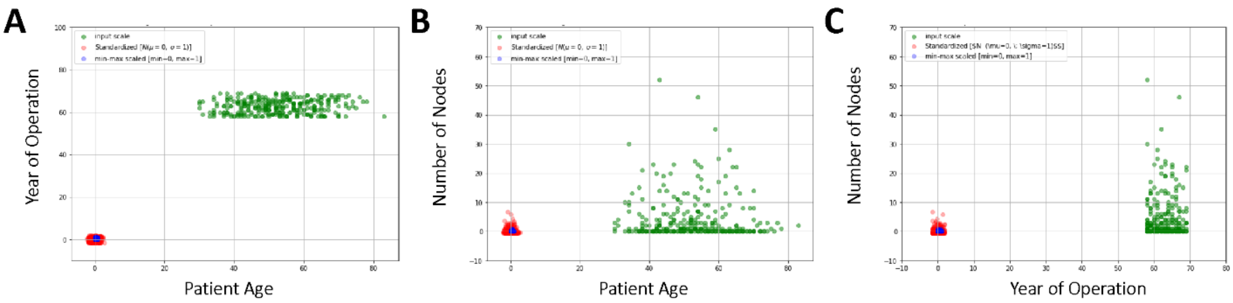


Fig. 7. Comparison of the distribution of breast cancer data before (green) and after standardization (red) or normalization (blue) of the data to compensate for differences in the measurement range for each feature. Plots of patient age versus **(A)** year of operation and **(B)** number of auxiliary nodes and year of operation versus number of auxiliary nodes.

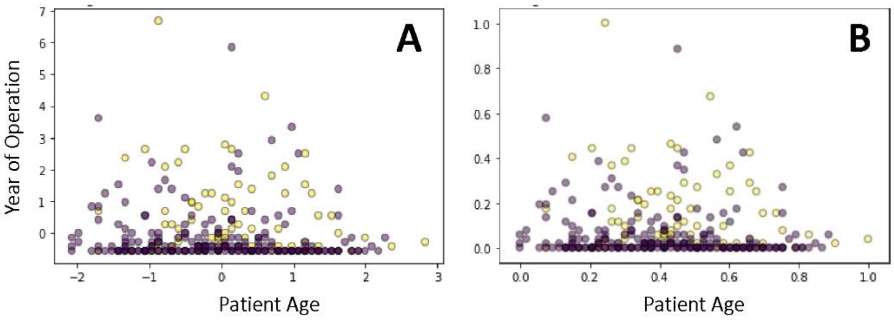


Fig. 8. Comparison of **(A)** standardized and **(B)** normalized data for the breast cancer dataset.

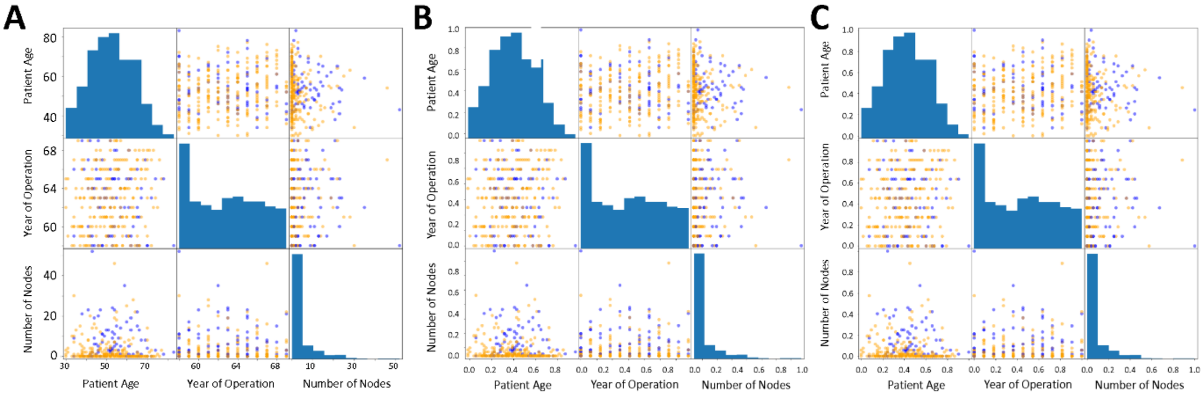


Fig. 9. Scatter plots for the **(A)** original, **(B)** standardized and **(C)** normalized breast cancer data.

Prediction Modelling

Support Vector Machine (SVM) Classifier

Hyperparameters for the SVM classifier were optimized using a systematic grid search. Table 5 shows the results for the prediction performance for the SVM classifier using the optimized hyperparameters (i.e. RBF kernel, gamma and C values of 0.001 and 1000, respectively). The classifier achieved a reasonable accuracy of about 69% although precision and recall were poor (i.e. 47% and 49% respectively) **(Table 5)**.

Table 5. Performance scores for the best combinations of hyperparameters obtained for the SVM classifier for the prediction of high-risk breast cancer patients.

Optimal Hyperparameters			**Classifier score (%)**			
C	Kernel	Gamma	Precision	Recall	F1-score	Accuracy
1000	rbf	0.001	46.61	48.68	45.36	68.85

k-Nearest Neighbour (kNN) Classifier

Hyperparameters influencing the performance of the *k*NN classifier were searched simultaneously via a grid search methodology using the GridSearchCV algorithm (Boschetti and Massaron, 2015). Initially, the whole standardized dataset was randomly divided into two series, a learning and validation series, to train and validate the final models using a ratio of 1:1. Five-fold cross-validation was used within the learning sample, which was randomly divided into five learning partitions of roughly equal size. The model was trained on four of these partitions and performance was estimated using the last partition (the test partition). The five estimates of performance were then averaged to obtain the cross-validation performance for the model.

The *k*NN algorithm is a non-parametric statistical model based on different combinations of the descriptive features (i.e. patient age, year of operation and number of positive auxiliary nodes). Model performance was measured by the ability of the model to predict class target labels (survival for 5 years or longer versus less than 5 years) pre-defined in the test portion of the dataset.

For each combination of descriptive features, using either the Euclidean or Manhattan method to measure the distance between nearest neighbours had negligible effect upon the performance of the model (**Appendix Tables 1-4**). Model performance was optimal for a *k* (nearest neighbours) value of 2 (prediction accuracy of 83.41%).

The accuracy performance score provides a measure of the ability of the *k*NN classifier to correctly predict target labels. This includes patients with a good prognosis as well as high risk patients with poor survival outcomes. It is likely that clinicians may be more interested in identifying individuals with a low chance of survival so that they can be prioritized for early therapy. To gain a more detailed understanding of how the model performed when predicting these high risk breast cancer patients, we determined the confusion matrix and performance scores for the model when systematically varying each hyperparameter. The performance *k*NN model was tested against the original 'un-processed', cube root transformed, normalized and standardized datasets (**Appendix Tables 5-16**).

Examination of the confusion matrices for each dataset revealed that the *k*NN classifier performed poorly when predicting high risk breast cancer patients. In the best-case scenario, the classifier could only predict 50% of the high-risk patients.

Overall, the best performance for the *k*NN classifier was obtained using a k (nearest neighbours) value of 4 and the euclidean distance measure with precision and recall scores of 0.55 and 0.50, respectively (**Table 6**).

Table 6. Performance scores for the best combinations of hyperparameters obtained for the *k*NN model for the prediction of high-risk breast cancer patients.

Dataset	K value	P value	Precision	Recall	F1-score	Model score (SCV)	Prediction of high risk patients (%)
Cube root transformed	3	2	0.44	0.50	0.47	0.611	50
Cube root transformed	4	2	0.55	0.50	0.50	0.618	50
Normalized	1	1	0.40	0.50	0.44	0.650	50
Normalized	1	2	0.41	0.44	0.42	0.660	44
Standardized	1	2	0.41	0.44	0.42	0.660	44

Decision Tree Classifier

We created two trees using default parameters and parameters obtained utilizing the GridSearchCV algorithm to examine the effects of overfitting. The grid search method was used to determine the hyperparameters that would give the best recall score.

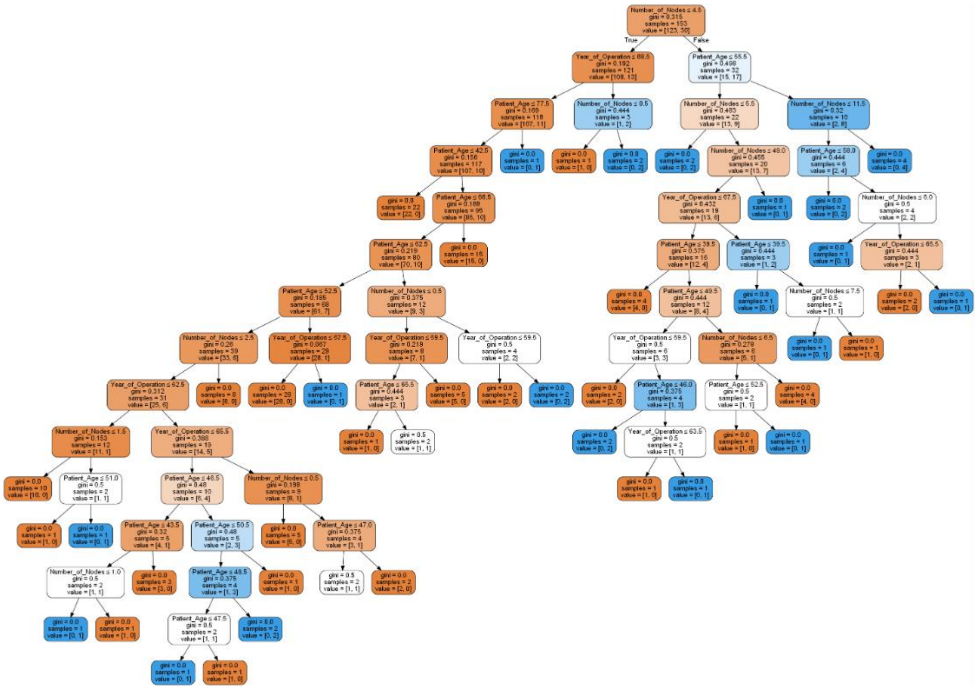
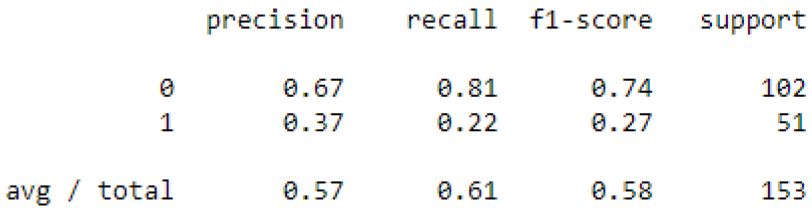


Fig. 10 Decision tree obtained using the default hyperparameters (max_depth=None, min_samples_split=2, min_samples_leaf=1, max_leaf_nodes=None).

Table 7. Performance scores for the decision tree generate using the default hyperparameters listed in **Fig. 10**.



```
             precision    recall  f1-score   support

          0       0.67      0.81      0.74       102
          1       0.37      0.22      0.27        51

avg / total       0.57      0.61      0.58       153
```

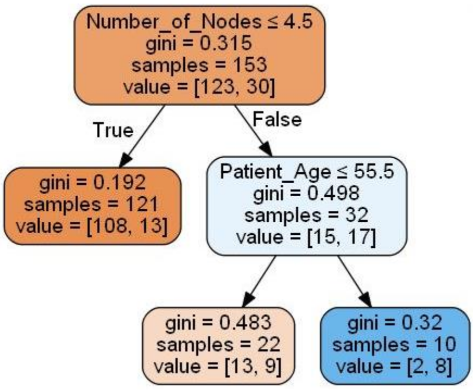
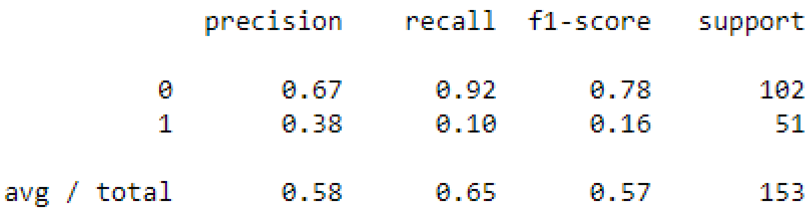


Fig. 11 Best performing decision tree generated using the optimized hyperparameters (max_depth=2, min_samples_split=5, min_samples_leaf=1, max_leaf_nodes=3).

Table 8. Performance scores for the decision tree generate using the optimized hyperparameters listed in **Fig. 11**.



```
             precision    recall  f1-score   support

          0       0.67      0.92      0.78       102
          1       0.38      0.10      0.16        51

avg / total       0.58      0.65      0.57       153
```

Introducing constraints upon the decision tree model drastically reduced its overall complexity while minimizing overfitting (**Figs 10 & 11**). This resulted in an increase in the model's performance (**Tables 7 & 8**). However, from a clinical perspective we would likely be more interested identifying patients having poor survival prospects. In which case the optimized tree, with a recall score of 10% for target=1 (high risk patients), performs worse than the overfitted tree, with a recall score of 22% for target=1.

Discussion

The SVM classifier is a powerful and advanced supervised learning method that has been reported to work well with small data sets (Huang et al., 2017). We have found that the SVM classifier performed reasonably well at predicting breast cancer patients with a high likelihood of surviving beyond 5 years but poorly when attempting to identify high risk patients with a low chance of survival (Table 8).

The *k*NN classifier makes predictions based on the k nearest neighbours and is a very robust classifier. The hyperparameters *k* (nearest neighbours), distance metric (manhattan or Euclidean) and weighting (uniform or distance) can be varied to optimize the performance of the model. The distance metric determines how the distance is measured between a point and other points in the dataset. The weighting can be applied to weight closer points more highly giving them greater importance.

The Decision Tree classifier was useful because the ability to see the visualization of the tree really helps to understand what the classification algorithm is doing. It also helps to decide which parameters to tune to improve the classification. An example would be if the gini index is not improving from one level of the tree to the next, it may help to reduce the maximum depth of the tree to avoid growing the tree unnecessarily.

Decision tree model has the benefit of having more variables which can be tuned to fit the data compared to the SVM and *k*NN classifiers. However, this also means that it can take significantly more time to optimize the hyperparameters compared to these classifiers even when using a grid search algorithm. It is possible that the recall score for the decision tree may have been improved if more values were added to the parameter grid for the grid search, however it would have exponentially increased the computational time required to optimize the parameters.

Overall, the SVM, *k*NN and decision tree classification models performed reasonable well at classifying breast cancer patients based on their survival status (**Table 9**). The SVM and *k*NN classifiers outperformed the decision tree model with classification accuracies of 74% and 69%, respectively (compared to 62% for the decision tree model). All models performed significantly better at identifying long term breast cancer survivors (recall values of 91%, 82% and 92% for the SVM, *k*NN and decision tree models, respectively) compared to the high risk breast cancer patients (recall values of 6.0%, 50% and 10% for the SVM, *k*NN and decision tree models, respectively). The SVM and decision tree classifiers performed particularly poorly at identifying high risk patients.

Table 9. Comparison of the of the SVM, *k*NN and decision tree classifiers for predicting breast cancer patients based on survival status.

Classification Model	**Survival < 5 years**			**Survival > 5 years**				
	Precision	Recall	F1-score	Precision	Recall	F1-score	Classification error rate	Classification accuracy
SVM	0.20	0.06	0.10	0.73	0.91	0.81		0.688
*k*NN	0.55	0.50	0.50	0.82	0.82	0.82	0.262	0.738
Decision Tree	0.38	0.10	0.16	0.67	0.92	0.78	0.381	0.619

Conclusions

The goal of this study was to develop a machine learning classification model for the identification of high risk breast cancer patients having a poor chance of long term survival, so that they may be targeted for high priority therapy. Overall, the three prediction models (SVM, *k*NN and decision tree classifiers) performed reasonably well at identify breast cancer patients based on their survival status but poorly when considering the more clinically relevant high risk patients.

References

BOSCHETTI , A. & MASSARON, L. 2015. *Python Data Science Essentials,* Birmingham, PACKT publishing.

FOUAD, T. M., KOGAWA, T., LIU, D. D., SHEN, Y., MASUDA, H., EL-ZEIN, R., WOODWARD, W. A., CHAVEZ-MACGREGOR, M., ALVAREZ, R. H., ARUN, B., LUCCI, A., KRISHNAMURTHY, S., BABIERA, G., BUCHHOLZ, T. A., VALERO, V. & UENO, N. T. 2015. Overall survival differences between patients with inflammatory and noninflammatory breast cancer presenting with distant metastasis at diagnosis. *Breast cancer research and treatment,* 152**,** 407-416.

FRIEDMAN, J., HASTIE, T. & TIBSHIRANI, R. 2001. *Elements of Statistical Learning: Data mining, inference, and prediction.,* New York, NY, Springer.

HABERMAN, S. J. 1976. Generalized residuals for log-linear models. *Proceedings of the 9th International Biometrics Conference.* Boston.

HARBECK, N. & GNANT, M. 2017. Breast cancer. *Lancet (London, England),* 389**,** 1134-1150.

HUANG, M.-W. W., CHEN, C.-W. W., LIN, W.-C. C., KE, S.-W. W. & TSAI, C.-F. F. 2017. SVM and SVM Ensembles in Breast Cancer Prediction. *PloS one,* 12.

KELLEHER, J., MAC NAMEE, B. & D'ARCY, A. 2015. *Fundamentals of Machine Learning for Predictive Data Analytics: Algorithms, Worked Examples, and Case Studies*.

PEART, O. 2017. Metastatic Breast Cancer. *Radiologic technology,* 88.

PEDREGOSA, F., VAROQUAUX, G., GRAMFORT, A., MICHEL, V., THIRION, B., GRISEL, O., BLONDEL, M., PRETTENHOFER, P., WEISS, R., DUBOURG, V., VANDERPLAS, J., PASSOS, A., COURNAPEAU, D., BRUCHER, M., PERROT, M. & DUCHESNAY, É. 2011. Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research,* 12**,** 2825-2830.

SIEGEL, R. L., MILLER, K. D. & JEMAL, A. 2016. Cancer statistics, 2016. *CA: a cancer journal for clinicians,* 66**,** 7-30.

Appendix

Table 1. Hyperparameter optimization for nearest neighbours using a grid search approach while keeping the distance metric (i.e. euclidean distance) and weighting (i.e. uniform) constant.

k value	Distance Metric	Weights	Accuracy (%)	Optimal neighbours
1, 2, 3, 4, 5, 6, 7, 10	Euclidean	Uniform	78.17	10
1, 2, 3, 4, 5, 6, 7	Euclidean	Uniform	77.29	7
1, 2, 3, 4, 5, 6	Euclidean	Uniform	83.41	2
1, 2, 3, 4, 5	Euclidean	Uniform	83.41	2
1, 2, 3, 4	Euclidean	Uniform	83.41	2
1, 2, 3	Euclidean	Uniform	83.41	2

Table 2. Hyperparameter optimization for nearest neighbours using a grid search approach while keeping the distance metric (i.e. manhattan distance) and weighting (i.e. uniform) constant.

k value	Distance Metric	Weights	Accuracy (%)	Optimal neighbours
1, 2, 3, 4, 5, 6, 7, 10	Manhattan	Uniform	78.17	10
1, 2, 3, 4, 5, 6, 7	Manhattan	Uniform	77.29	7
1, 2, 3, 4, 5, 6	Manhattan	Uniform	83.41	2
1, 2, 3, 4, 5	Manhattan	Uniform	83.41	2
1, 2, 3, 4	Manhattan	Uniform	83.41	2
1, 2, 3	Manhattan	Uniform	83.41	2

Table 3. Hyperparameter optimization for nearest neighbours using a grid search approach while keeping the distance metric (i.e. euclidean distance) and weighting (i.e. distance) constant.

k value	Distance Metric	Weights	Accuracy (%)	Optimal neighbours
1, 2, 3, 4, 5, 6, 7, 10	Euclidean	Distance	78.17	10
1, 2, 3, 4, 5, 6, 7	Euclidean	Distance	77.29	7
1, 2, 3, 4, 5, 6	Euclidean	Distance	83.41	2
1, 2, 3, 4, 5	Euclidean	Distance	83.41	2
1, 2, 3, 4	Euclidean	Distance	83.41	2
1, 2, 3	Euclidean	Distance	83.41	2

Table 4. Hyperparameter optimization for nearest neighbours using a grid search approach while keeping the distance metric (i.e. manhattan distance) and weighting (i.e. distance) constant.

k value	Distance Metric	Weights	Accuracy (%)	Optimal neighbours
1, 2, 3, 4, 5, 6, 7, 10	Manhattan	Distance	78.17	10
1, 2, 3, 4, 5, 6, 7	Manhattan	Distance	77.29	7
1, 2, 3, 4, 5, 6	Manhattan	Distance	83.41	2
1, 2, 3, 4, 5	Manhattan	Distance	83.41	2
1, 2, 3, 4	Manhattan	Distance	83.41	2
1, 2, 3	Manhattan	Distance	83.41	2

Table 5. Confusion Matrix for *k*NN classifier predictions using the original 'un-processed' dataset. Correct (true) predictions are highlighted in red while incorrect predictions are black (weights = distance).

					k value			
p value[a]	1	2	3	4	5	6	7	10
1	42 3 13 3	42 3 14 2	43 2 13 3	42 3 13 3	42 3 13 3	43 2 13 3	42 3 13 3	42 3 13 3
2	43 2 13 3	43 2 13 3	42 3 13 3	41 4 13 3	42 3 13 3	41 4 13 3	42 3 13 3	42 3 13 3

[a]Distance metric (p = 1 manhattan distance; p=2 euclidean distance)

Table 6. Classifier performance scores for *k*NN classifier predictions for manhattan distance metric using the original 'un-processed' dataset. Correct predictions are highlighted in red while incorrect predictions are black (weights = distance).

				k value				
	1	2	3	4	5	6	7	10
Precision	0.82 0.29	0.75 0.4	0.77 0.60	0.76 0.50	0.76 0.50	0.77 0.60	0.76 0.50	0.76 0.50
Recall	0.67 0.47	0.93 0.12	0.96 0.19	0.93 0.19	0.93 0.19	0.96 0.19	0.93 0.19	0.93 0.19
F1-score	0.74 0.36	0.83 0.19	0.85 0.29	0.84 0.27	0.84 0.27	0.85 0.29	0.84 0.27	0.84 0.27

Model score (5CV)[a]	0.74828	0.69968	0.69318	0.67705	0.70597	0.741830	0.72221	0.72538

[a] 5CV – 5-fold cross-validation

Table 7. Classifier performance scores for *k*NN classifier predictions for euclidean distance metric using the original 'un-processed' dataset. Correct predictions are highlighted in red while incorrect predictions are black (weights = distance).

				K value				
	1	2	3	4	5	6	7	10
Precision	0.77 0.60	0.77 0.60	0.76 0.50	0.76 0.43	0.76 0.50	0.76 0.43	0.76 0.50	0.76 0.50
Recall	0.96 0.19	0.96 0.19	0.93 0.19	0.91 0.19	0.93 0.19	0.91 019	0.93 0.19	0.93 0.19
F1-score	0.85 0.29	0.85 0.29	0.84 0.27	0.83 0.26	0.84 0.27	0.83 0.26	0.84 0.27	0.84 0.27
Model score (5CV)	0.68780	0.69935	0.68667	0.68990	0.74828	0.74167	0.74162	0.73834

Table 8. Confusion Matrix for *k*NN classifier predictions using the original dataset but where the auxiliary nodes data has been processed with a cubed root function. Correct (true) predictions are highlighted in red while incorrect predictions are black (weights = distance).

					k value			
p value[a]	1	2	3	4	5	6	7	10
1	39 9 10 6	39 6 10 6	40 5 12 4	41 4 11 5	44 1 15 1	45 0 15 1	43 2 14 2	45 0 15 1
2	40 5 10 6	40 5 10 6	35 10 8 8	37 8 8 8	43 2 13 3	45 0 14 2	45 1 15 1	45 0 16 0

[a] Distance metric (p = 1 manhattan distance; p=2 euclidean distance)

Table 9. Classifier performance scores for *k*NN classifier predictions for manhattan distance metric using the original dataset but where the auxiliary nodes data has been processed with a cubed root function. Correct predictions are highlighted in red while incorrect predictions are black (weights = distance).

				k value				
	1	2	3	4	5	6	7	10
Precision	0.78	0.80	0.77	0.76	0.75	0.75	0.75	0.75

	0.40	0.50	0.44	0.56	0.50	1.00	0.50	1.00
Recall	0.80 0.38	0.87 0.38	0.89 0.25	0.91 0.31	0.98 0.06	1.00 0.06	0.96 0.12	1.00 0.06
F1-score	0.79 0.39	0.83 0.43	0.82 0.32	0.85 0.40	0.85 0.11	0.86 0.12	0.84 0.20	0.86 0.12
Model score (5CV)[a]	0.63723	0.67652	0.64093	0.67023	0.68646	0.68969	0.69603	0.71866

[a]5CV – 5-fold cross-validation

Table 10. Classifier performance scores for *k*NN classifier predictions for euclidean distance metric using the original dataset but where the auxiliary nodes data has been processed with a cubed root function. Correct predictions are highlighted in red while incorrect predictions are black (weights = distance).

				K value				
	1	2	3	4	5	6	7	10
Precision	0.80 0.55	0.80 0.55	0.81 0.44	0.82 0.50	0.77 0.60	0.76 1.00	0.75 0.50	0.74 0.00
Recall	0.89 0.38	0.89 0.38	0.78 0.50	0.82 0.50	0.96 0.19	1.00 0.12	0.98 .06	1.00 0.00
F1-score	0.84 0.44	0.84 0.44	0.80 0.47	0.82 0.50	0.85 0.29	0.87 0.22	0.85 0.11	0.85 0.00
Model score (5CV)[a]	0.60471	0.61782	0.61148	0.61798	0.6832	0.68302	0.70549	0.70544

[a]5CV – 5-fold cross-validation

Table 11. Confusion Matrix for *k*NN classifier predictions using the normalized dataset. Correct (true) predictions are highlighted in red while incorrect predictions are black (weights = distance).

					k value			
p value[a]	1	2	3	4	5	6	7	10
1	36 9 10 6	38 7 12 4	40 5 11 5	38 7 12 4	41 4 13 3	41 4 13 3	41 4 13 3	41 4 13 3
2	35 10 9 7	40 5 14 2	42 3 15 1	41 4 14 2	43 2 13 3	40 5 13 3	42 3 13 3	41 4 13 3

[a]Distance metric (p = 1 manhattan distance; p=2 euclidean distance)

Table 12. Classifier performance scores for *k*NN classifier predictions for manhattan distance metric using the normalized dataset. Correct predictions are highlighted in red while incorrect predictions are black (weights = distance).

				k value				
	1	2	3	4	5	6	7	10
Precision	0.78 0.40	0.76 0.36	0.78 0.50	0.76 0.36	0.76 0.43	0.76 0.43	0.76 0.43	0.76 0.43
Recall	0.80 0.38	0.84 0.25	0.89 0.31	0.84 0.25	0.91 0.19	0.91 0.19	0.91 0.19	0.91 0.19
F1-score	0.79 0.39	0.80 0.30	0.83 0.38	0.80 0.30	0.83 0.26	0.83 0.26	0.83 0.26	0.83 0.26
Model score (5CV)[a]	0.65045	0.66028	0.66383	0.65404	0.71253	0.69947	0.69926	0.71560

[a]5CV – 5-fold cross-validation

Table 13. Classifier performance scores for *k*NN classifier predictions for euclidean distance metric using the normalized dataset. Correct predictions are highlighted in red while incorrect predictions are black (weights = distance).

				k value				
	1	2	3	4	5	6	7	10
Precision	0.80 0.41	0.74 0.29	0.74 0.25	0.75 0.33	0.77 0.60	0.74 0.38	0.76 0.50	0.76 0.43
Recall	0.78 0.44	0.89 0.12	0.93 0.06	0.91 0.12	0.96 0.19	0.89 0.19	0.93 0.19	0.91 0.19
F1-score	0.79 0.42	0.81 0.17	0.82 0.10	0.82 0.18	0.85 0.29	0.82 0.25	0.84 0.27	0.83 0.26
Model score (5CV)[a]	0.66039	0.66695	0.67372	0.68339	0.71243	0.70582	0.72216	0.74172

[a]5CV – 5-fold cross-validation

Table 14. Confusion Matrix for *k*NN classifier predictions using the standardized dataset. Correct (true) predictions are highlighted in red while incorrect predictions are black (weights = uniform).

					k value			
p value	1	2	3	4	5	6	7	10

1	33 12 8 8	41 4 13 3	39 6 11 5	42 3 15 1	41 4 13 3	41 4 13 3	41 4 13 3	44 1 14 2
2	35 10 9 7	44 1 15 1	41 4 15 1	44 1 15 1	42 3 13 3	43 2 13 3	43 2 13 3	42 3 14 2

Distance metric (p = 1 manhattan distance; p=2 euclidean distance)

Table 15. Classifier performance scores for *k*NN classifier predictions for manhattan distance metric using the standardized dataset. Correct predictions are highlighted in red while incorrect predictions are black (weights = uniform).

				k value				
	1	2	3	4	5	6	7	10
Precision	0.80 0.40	0.76 0.43	0.78 0.45	0.74 0.25	0.76 0.43	0.76 0.43	0.76 0.43	0.76 0.67
Recall	0.73 0.50	0.91 0.19	0.87 0.31	0.93 0.06	0.91 0.19	0.91 0.19	0.91 0.19	0.98 0.12
F1-score	0.77 0.44	0.83 0.26	0.82 0.37	0.82 0.10	0.83 0.26	0.83 0.26	0.83 0.26	0.85 0.21
Model score (5CV)[a]	0.65045	0.69963	0.65420	0.70910	0.72226	0.71238	0.69937	0.71872

[a]5CV – 5-fold cross-validation

Table 16. Classifier performance scores for *k*NN classifier predictions for euclidean distance metric using the standardized dataset. Correct predictions are highlighted in red while incorrect predictions are black (weights = uniform).

				K value				
	1	2	3	4	5	6	7	10
Precision	0.80 0.41	0.754 0.50	0.73 0.20	0.75 0.50	0.76 0.50	0.77 0.60	0.77 0.60	0.75 0.40
Recall	0.78 0.44	0.98 0.06	0.91 0.06	0.98 0.06	0.93 0.19	0.96 0.19	0.96 0.19	0.93 0.12
F1-score	0.79 0.42	0.85 0.11	0.81 0.10	0.85 0.11	0.84 0.27	0.85 0.29	0.85 0.29	0.83 0.19
Model score (5CV)[a]	0.66039	0.70920	0.66721	0.74166	0.71243	0.72538	0.72872	0.73501

[a] 5CV – 5-fold cross-validation