Machine Learning Algorithms for Predicting Breast Cancer Survival Status.

Harith Palakal Mavilavalapil(ID-14197170) Vivek Kalarikkal Subramanian (ID- 13163323)

*MSc Data Science, Coventry University, Coventry, UK*

This paper explores the **Chances of Survival for those with Breast cancer**. The survival rate has been predicted; by analysing their behaviour and applying various machine learning algorithms such as Decision Tree, KNN, SVM & Logistic Regression. To prepare the data, to address class imbalance issues, categorical data encoding, and feature extraction, multiple techniques are used. Python programming language has been used to perform the techniques and algorithms. Machine learning repositories were used to obtain prediction results and algorithm performance measures. These results then visualized for comparison and discussion.

|  |  |  |  |
| --- | --- | --- | --- |
| 7 | Differenti ate | Categorical | Moderately differentiated. Poorly differentiated, Undifferentiated, Well  differentiated |
| 8 | Grade | Categorical | anaplastic; Grade IV,1,2,3 |
| 9 | A Stage | Categorical | Distant, Regional |
| 10 | Tumour Size | Numerical |  |
| 11 | Estrogen Status | Categorical | Positive or Negative |
| 12 | Progester one Status | Categorical | Positive or Negative |
| 13 | Regional Node Examined | Numerical |  |
| 14 | Regional Node Positive | Numerical |  |
| 15 | Survival Months | Numerical |  |
| 16 | Status | Categorical | Alive or Dead |

***Keywords—machine learning; python; categorical data; Breast cancer survival, , k-nearest, support vector machines, Logistic Regression, Decision Tree.***

##### INTRODUCTION

Breast cancer is a major concern for women globally, though it can rarely affect men. It starts in breast tissue, often in milk-producing glands or milk ducts, and forms malignant tumours. Early detection through mammograms has significantly boosted survival rates. Treatment options like surgery, chemotherapy, radiation, or hormone therapy are opted based on cancer stage, type, and the patient's health. ML helps to predict survival rates by analysing patient’s characteristics and clinical factors to estimate survival chances after a breast cancer diagnosis. This paper emphasizes the positive impact of early detection.

##### THE DATA SET

The dataset obtained from Kaggle and consists of 4024 instances. Each instance represents one person's medical status visit to the site, containing 16 attributes with no missing values. The first fourteen attributes describe medical details and consist of 6 numerical values and 11 categorical features. The last attribute represents the survival status of the patient.

Table 1 displays a summary of the dataset, describing the features, their types and value ranges for categorical features.

|  |  |  |  |
| --- | --- | --- | --- |
| NO | Descriptio n | Type | Categorical Value Range |
| 1 | Age | Numeric |  |
| 2 | Race | Categorical | White/Black/Others |
| 3 | Marital Status | Categorical | Single, Married, Separated & widowed |
| 4 | T Stage | Categorical | T1-T4 |
| 5 | N Stage | Categorical | N1-N3 |
| 6 | 6th Stage | Categorical | IIA, IIIA, IIB, IIIC, IIIB |

Table 1. Dataset Features

##### DATA PREPARATION

This portion of the content focuses on examining data, managing and processing categorical data, encoding categorical data, scrutinizing numerical values and scaling data, conducting feature analysis, and extracting features from the data.

###### Categorical Feature Encoding

Although machine learning algorithms require numerical values, categorical data can be represented as text. However, chances of failing are higher in ML, even if even if the categorical data is represent as a number. To avoid this situation, categorical data must transform into binary values. Additionally, to simplify the dataset and lower its dimensionality, dummy variables should eliminate. In this research paper, 10 categorical features have been encoded. Binary vectors were created from category variables using one-hot encoding technique. Every category has its own column, and in each column, a value of 1 & 0 denotes the category's presence and absence respectively (Seger, 2018) Since the algorithm treats each category equally and does not assume an ordinal relationship between them, this method is especially well- suited for categorical variables with no intrinsic order among the categories.

##### MACHINE LEARNING CLASSIFICATION METHODS

###### DECISION TREE

A **Tree structure** represents a classification model, which aims to split a dataset into smaller subsets based on the similarities between observations. To access these subsets, specific decisions are required. The algorithm used to generate the tree, such as ID3, calculates the information gain to determine how well the records are separated into the target variable using a given attribute. Highest information gain attribute is selected to calculate the amount of entropy represented by the attribute.

𝐸𝑛𝑡𝑟𝑜𝑝𝑦 (𝑆)=Σ𝑝𝑖(𝑙𝑜𝑔2𝑝𝑖)

The decision tree algorithm is a useful tool because of its straightforwardness in understanding and visualization. Another significant advantage is that it doesn't require any data preparation techniques. However, the algorithm doesn't support empty values, which are removed before the classification process. On the other hand, when the data is too complicated, an excessively complex tree may lead to overfitting and inaccurate predictions. (Rokach, 2008).

###### K-NEAREST NEIGHBOURS (KNN)

The **K-Nearest Neighbours** algorithm is a classification method that is both simple to comprehend and apply. KNN is a potent tool that can deliver precise predictions. It is particularly useful in models with only a small number of attributes that determine the class, making it a viable option for various applications.

In essence, KNN operates by computing the distance between each instance and allocating the unclassified data to its closest class (neighbourhood). This is achieved through a distance function, such as Euclidean distance, and the data is subsequently classified based on the class of its nearest neighbours.

The first step in using the K-Nearest Neighbours (KNN) algorithm is figuring out the value of K, which stands for the total number of nearest neighbours taken into account. Next, using distance metrics like Manhattan or Euclidean distances, it calculates the distances between each unclassified data point and every other point in the training dataset. Based on these distances, the programme then determines who the K closest neighbours are to the undeclared site. It locates the closest neighbours and, based on a majority vote among the neighbours' class labels, provides a class label to the unclassified location. KNN is a flexible classification method that can be used to solve challenging issues where conventional parametric models find it difficult to precisely specify decision limits (Rajaguru & Prabhakar, 2017)

###### SUPPORT VECTOR MACHINES (SVM)

For problems involving regression and classification, the supervised learning algorithm Support Vector Machine (SVM) is employed. When working with high-dimensional data or when the data cannot be separated linearly, it is especially helpful. A decision boundary called a hyperplane that separates the feature space into areas that correspond to various classes. (Dayananda, 2023). The hyperplane that maximises the margin the distance between the hyperplane and the closest support vector data points is the ideal one for each class.

SVM does this by employing a kernel function to translate the input data into a higher-dimensional space. The data points become more separable in this higher-dimensional space, which facilitates the identification of the ideal hyperplane. SVM can be applied with a variety of kernel functions, including radial basis function (RBF), polynomial, and linear. The type of data and the issue at hand determine which kernel function is best (Steinwart & Christmann, 2008).

###### LOGISTIC REGRESSION

Logistic regression is a statistical technique used for binary classification tasks, with the goal of predicting the likelihood of an observation falling into one of two classes based on input features. Unlike linear regression, which predicts continuous outcomes. Logistic regression models the probability of a specific event occurring. LR uses a logistic function (aka sigmoid function) to linearly combine input features and coefficients. This function converts the output into a 0–1 range, representing the probability that the observation belongs to the positive class.

Regularisation of the logistic regression model can lessen overfitting and enhance its generalisation capabilities. A common regularisation method is the L1 penalty, sometimes referred to as LASSO (Least Absolute Shrinkage and Selection Operator). The regression coefficients' absolute values are added via the L1 penalty to the objective function that is being optimised. Large coefficients are penalised by this new restriction, which encourages some of them to converge to zero. Consequently, certain features can be practically eliminated from the model due to their zero coefficients (Boateng & Abaye, 2019).

##### EXPERIMENTAL SETUP

An HP laptop equipped with an Intel Core i7-6700HQ CPU and 8GB RAM was used to train the models. Classification algorithms were implemented using the scikit-learn library. The dataset was cleaned, and feature extraction was done before encoding. Further, the dataset was divided into training and testing sets using **train\_test\_split**. Four classification algorithms were used to assess performance and accuracy.

##### APPLICATION OF TECHNIQUES AND RESULTS

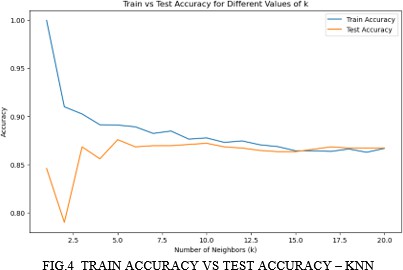
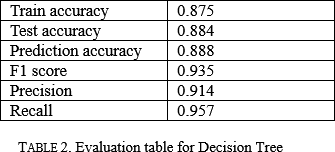
The models have been trained using the best parameters that were obtained through a grid search. For each model, its performance was evaluated by obtaining a confusion matrix, ROC and model metrics.

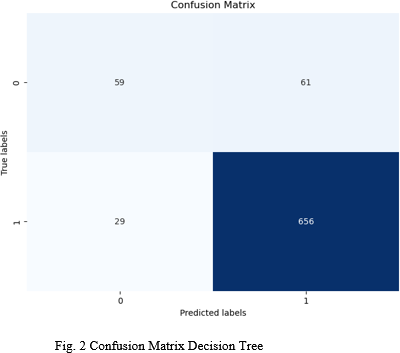
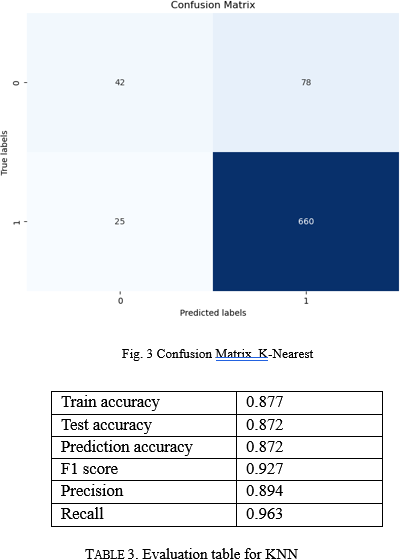
To get started with the required tasks, we need to import several scientific libraries that contain pre-defined functions. For the main part of the project, which involves classification algorithms, the scikit-learn library has been used.

###### A graph of a graph with blue and orange lines Description automatically generatedA DECISION TREE

The decision tree demonstrates exceptional performance across several metrics. Notably, its high accuracy (around 88%) on both training and test data indicates effective classification of a substantial portion of your data. This strength is further bolstered by the impressive F1-score (0.935), suggesting the model excels at identifying both positive and negative cases. This balance is further reflected in the high precision (0.914) for accurate positive predictions and the strong recall (0.957) for capturing most positive instances. While the slightly higher recall might point towards a bias favouring positive classifications, the overall performance suggests good generalization. Additionally, the decision tree performs particularly well for positive cases, evidenced by the high number of true positives and relatively low number of false negatives.

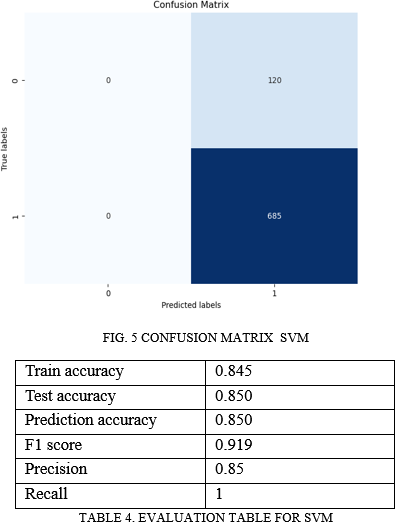
###### B K-NEAREST NEIGHBOURS (KNN)



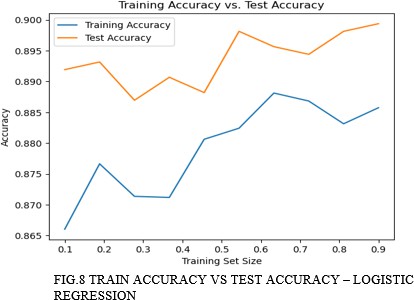
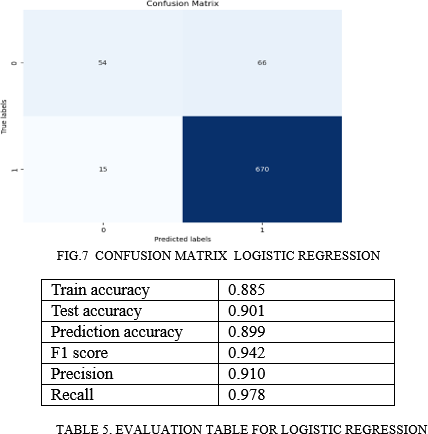
We used StandardScaler to verify that all features have the same scale, which is critical for KNN models. Prior to this, the model indicated potential overfitting of the data. After changing the K number and feature calling, we got more accurate results than the previous ones. For standardizing features, we performed standard scalar function to scale the data to avoid over fitting issues (Brownlee, 2020). The KNN model exhibits good performance, especially for identifying positive cases (high true positives, low false negatives), but might favour positive classifications (more false positives). The KNN model exhibits promising performance with a strong F1-score (0.927), indicating a good balance between precision (0.894) and recall (0.963) for accurate classifications. The high-test accuracy (0.872) suggests good generalizability.

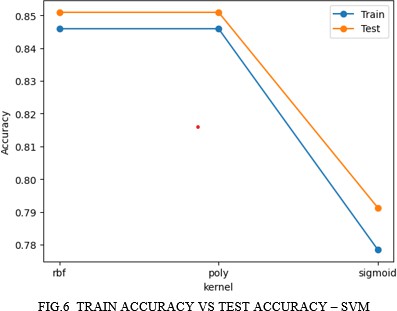
###### C. SUPPORT VECTOR MACHINES (SVM)



The SVM model excels with high performance (test accuracy: 0.850, F1-score: 0.919) and a balanced precision-recall (0.85, 1). The smallest train-test difference (0.005) indicates good generalisation. While recall excels at detecting positives, a larger dataset could confirm these findings and investigate hyperparameter tuning for additional optimisation. The model does well in both positive and negative classes, with a large number of true positives and true negatives.

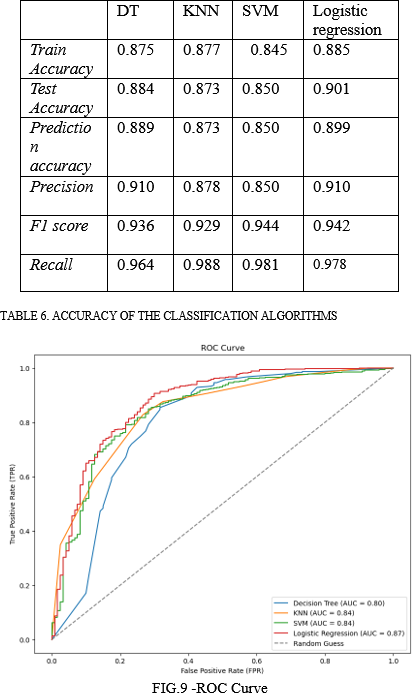
###### E. LOGISTIC REGRESSION



The linear regression model excels with a high test accuracy (0.901) and a great F1-score (0.942), showing that it captures the linear relationship between variables and generalizes well.

Balanced precision (0.910) and recall (0.978) indicate accurate predictions with few errors. The logistic regression model performs well based on the confusion matrix, particularly for the positive class. However, the presence of false positives indicates there is space for development.

1. **Discussion and Conclusions**



The ROC curves and AUC values for logistic regression, KNN, and SVM provide valuable insights into their classification performance:

* **Logistic Regression:** With the highest AUC of 0.87, logistic regression emerges as the strongest performer among the three models. Its ROC curve demonstrates a good balance between identifying true positive and avoiding false positive classifications (Omar & Ivrissimtzis, 2019).
* **Support Vector Machine (SVM):** While trailing slightly behind logistic regression with an AUC of 0.84, SVM shows promising performance. Its ROC curve also suggests a good balance between true positives and false positives.
* **K-Nearest Neighbours (KNN):** Despite a respectable AUC of 0.84, the decision tree falls behind logistic regression. This suggests it might be less effective at differentiating between positive and negative cases. However, decision trees offer the advantage of interpretability, which could be crucial if understanding the model's reasoning is important.
* **Decision Tree:** exhibits the lowest AUC (0.80) among the three models, indicating weaker overall performance in distinguishing between positive and negative classes.

**Overall:** Logistic regression appears to be the most suitable model for this classification task based on the AUC values.

#### REFERENCES

1. Alpaydin, E. (2020). Introduction to Machine Learning, fourth edition. In *Google Books*. MIT Press. [https://books.google.co.uk/books?hl=en&lr=&id=tZnSD](https://books.google.co.uk/books?hl=en&lr&id=tZnSDwAAQBAJ&oi=fnd&pg=PR7&dq=Introduction%2Bto%2BMachine%2BLearning%2B(3rd%2Bed.).&ots=F4SQ6UauDl&sig=5xCJT6UBrMOqIEoBzojRgwUuE8k%23v%3Donepage&q=Introduction%20to%20Machine%20Learning%20(3rd%20ed.).&f=false) [wAAQBAJ&oi=fnd&pg=PR7&dq=Introduction+to+Ma](https://books.google.co.uk/books?hl=en&lr&id=tZnSDwAAQBAJ&oi=fnd&pg=PR7&dq=Introduction%2Bto%2BMachine%2BLearning%2B(3rd%2Bed.).&ots=F4SQ6UauDl&sig=5xCJT6UBrMOqIEoBzojRgwUuE8k%23v%3Donepage&q=Introduction%20to%20Machine%20Learning%20(3rd%20ed.).&f=false) [chine+Learning+(3rd+ed.).&ots=F4SQ6UauDl&sig=5xC](https://books.google.co.uk/books?hl=en&lr&id=tZnSDwAAQBAJ&oi=fnd&pg=PR7&dq=Introduction%2Bto%2BMachine%2BLearning%2B(3rd%2Bed.).&ots=F4SQ6UauDl&sig=5xCJT6UBrMOqIEoBzojRgwUuE8k%23v%3Donepage&q=Introduction%20to%20Machine%20Learning%20(3rd%20ed.).&f=false) [JT6UBrMOqIEoBzojRgwUuE8k#v=onepage&q=Introdu](https://books.google.co.uk/books?hl=en&lr&id=tZnSDwAAQBAJ&oi=fnd&pg=PR7&dq=Introduction%2Bto%2BMachine%2BLearning%2B(3rd%2Bed.).&ots=F4SQ6UauDl&sig=5xCJT6UBrMOqIEoBzojRgwUuE8k%23v%3Donepage&q=Introduction%20to%20Machine%20Learning%20(3rd%20ed.).&f=false) [ction%20to%20Machine%20Learning%20(3rd%20ed.).](https://books.google.co.uk/books?hl=en&lr&id=tZnSDwAAQBAJ&oi=fnd&pg=PR7&dq=Introduction%2Bto%2BMachine%2BLearning%2B(3rd%2Bed.).&ots=F4SQ6UauDl&sig=5xCJT6UBrMOqIEoBzojRgwUuE8k%23v%3Donepage&q=Introduction%20to%20Machine%20Learning%20(3rd%20ed.).&f=false) [&f=false](https://books.google.co.uk/books?hl=en&lr&id=tZnSDwAAQBAJ&oi=fnd&pg=PR7&dq=Introduction%2Bto%2BMachine%2BLearning%2B(3rd%2Bed.).&ots=F4SQ6UauDl&sig=5xCJT6UBrMOqIEoBzojRgwUuE8k%23v%3Donepage&q=Introduction%20to%20Machine%20Learning%20(3rd%20ed.).&f=false)
2. Steinwart, I., & Christmann, A. (2008). Support Vector Machines. In *Google Books*. Springer Science & Business Media. [https://www.google.co.uk/books/edition/Support\_Vector](https://www.google.co.uk/books/edition/Support_Vector_Machines/HUnqnrpYt4IC?hl=en&gbpv=1&dq=svm&printsec=frontcover)

[\_Machines/HUnqnrpYt4IC?hl=en&gbpv=1&dq=svm&p](https://www.google.co.uk/books/edition/Support_Vector_Machines/HUnqnrpYt4IC?hl=en&gbpv=1&dq=svm&printsec=frontcover) [rintsec=frontcover.](https://www.google.co.uk/books/edition/Support_Vector_Machines/HUnqnrpYt4IC?hl=en&gbpv=1&dq=svm&printsec=frontcover)

1. Dayananda, S. (2023, November 6). Support Vector Machines (SVM). Medium. [https://sandundayananda.medium.com/support-vector-](https://sandundayananda.medium.com/support-vector-machines-svm-db8314e9092d#%3A~%3Atext%3DHyperplane(decision%20boundary)%26text%3DThe%20hyperplane%20separates%20the%20feature) [machines-svm-](https://sandundayananda.medium.com/support-vector-machines-svm-db8314e9092d#%3A~%3Atext%3DHyperplane(decision%20boundary)%26text%3DThe%20hyperplane%20separates%20the%20feature) [db8314e9092d#:~:text=Hyperplane(decision%20boundar](https://sandundayananda.medium.com/support-vector-machines-svm-db8314e9092d#%3A~%3Atext%3DHyperplane(decision%20boundary)%26text%3DThe%20hyperplane%20separates%20the%20feature) [y)&text=The%20hyperplane%20separates%20the%20fea](https://sandundayananda.medium.com/support-vector-machines-svm-db8314e9092d#%3A~%3Atext%3DHyperplane(decision%20boundary)%26text%3DThe%20hyperplane%20separates%20the%20feature) [ture](https://sandundayananda.medium.com/support-vector-machines-svm-db8314e9092d#%3A~%3Atext%3DHyperplane(decision%20boundary)%26text%3DThe%20hyperplane%20separates%20the%20feature)
2. Boateng, E. Y., & Abaye, D. A. (2019). A Review of the Logistic Regression Model with Emphasis on Medical Research. *Journal of Data Analysis and Information Processing*, *07*(04), 190–207.

<https://doi.org/10.4236/jdaip.2019.74012>

1. Omar, L., & Ivrissimtzis, I. (2019). Using theoretical ROC curves for analysing machine learning binary classifiers. *Pattern Recognition Letters*, *128*, 447–

451. <https://doi.org/10.1016/j.patrec.2019.10.004>

10. Brownlee, J. (2020, June 9). *How to Use StandardScaler and MinMaxScaler Transforms in Python*. Machine Learning Mastery.

[https://machinelearningmastery.com/standardscaler-and-](https://machinelearningmastery.com/standardscaler-and-minmaxscaler-transforms-in-python/)

1. Seger, C. (2018). *An investigation of categorical variable encoding techniques in machine learning: binary versus one-hot and feature hashing*. https://www.diva- portal.org/smash/get/diva2:1259073/FULLTEXT01.p

[minmaxscaler-transforms-in-python/](https://machinelearningmastery.com/standardscaler-and-minmaxscaler-transforms-in-python/)

#### DATA AVAILABILITY

<https://www.kaggle.com/datasets/reihanenamdari/breast-cancer>

df

1. Rokach, L. (2008). Data Mining with Decision Trees: Theory and Applications. In *Google Books*. World Scientific. [https://www.google.co.uk/books/edition/Data\_Minin](https://www.google.co.uk/books/edition/Data_Mining_with_Decision_Trees/GlKIIR78OxkC?hl=en&gbpv=0)

[g\_with\_Decision\_Trees/GlKIIR78OxkC?hl=en&gbp](https://www.google.co.uk/books/edition/Data_Mining_with_Decision_Trees/GlKIIR78OxkC?hl=en&gbpv=0)

[v=0](https://www.google.co.uk/books/edition/Data_Mining_with_Decision_Trees/GlKIIR78OxkC?hl=en&gbpv=0)

1. *sklearn.ensemble.AdaBoostClassifier — scikit-learn*

*0.22.1 documentation*. (n.d.). Scikit-Learn.org. [https://scikit-](https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.AdaBoostClassifier.html)

[learn.org/stable/modules/generated/sklearn.ensemble.](https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.AdaBoostClassifier.html)

[AdaBoostClassifier.html](https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.AdaBoostClassifier.html)

1. Brownlee, J. (2016). Machine Learning Algorithms From Scratch with Python. In *Google Books*. Machine Learning Mastery.