### COMP3009 MACHINE LEARNING ASSIGNMENT 2: Machine Learning for Disease Treatment Response Prediction

*Charlie Davies, Davlesh Jodhun, Kshitij Tiwari, Lithicka Anandavel, Rajat Goyal*

University of Nottingham

#### Abstract

Machine Learning (ML) can be used in medicine in order to predict outcomes for cancer patients after undergoing chemotherapy. With a wide range of models available, it is imperative that, for such an important implementation, the results are as accurate as possible. As such, various models need to be evaluated for this use. In particular, Support Vector Machines (SVM), Artificial Neural Networks (ANNs), Decision Trees (DTs), Random Forests (RFs) and Regression methods were examined. Ultimately, the SVM Classifier and the RF Regressor were chosen as the models to be used for final assessment.

**1. Introduction**

Breast cancer is the most common cancer in the UK, with a UK woman’s lifetime risk being 12.5% and is usually treated with a combination of surgery, chemotherapy and radiotherapy, as well as hormonal treatments, depending on the type of breast cancer [1]. Chemotherapy is typically used to reduce the size of the tumour before surgery. Chemotherapy, however, is toxic to the human body and is not always effective. Complete tumour resolution at surgery is known as Pathological Complete Response (PCR), which has a high likelihood of achieving cure and longer Relapse-Free Survival (RFS) time. RFS is the length of time that a patient survives for after primary treatment, without any further signs or symptoms of that cancer. Between 25-40% of patients who undergo chemotherapy will achieve a full PCR, while other patients may achieve near or partial PCR[[1]](#footnote-1) and the remaining 60-75% left with some residual disease and a range of prognosis [2].

This project will attempt to predict PCR and RFS using information prior to chemotherapy treatment, to achieve better patient stratification and better treatment overall. The models will be compared, and the best models will be chosen based mainly on accuracy, among other parameters. These models will then be tested on an unseen dataset for performance evaluation, which means that they must be optimised for generalisation.

**2. Background**

ML methods have increased in capability over the past decade, with increases in computational power matching the increase in the sheer amount of data being produced. The amount of data present in medical analysis can now be handled by ML, in particular predictions for disease and outcomes of treatment, such as the prediction of the progression from pre-diabetes to type 2 diabetes using electronic health data.

ML techniques use algorithms based on mathematical depictions of relations between variables, employing statistical tricks along the way. The relations between medical variables tend to be quite straightforward, for example, the relationship between ionising radiation exposure and cancer risk. The use of ML methods tends to answer the “what” questions about these relationships between data, since the primary concern is being able to accurately predict outcomes [3] over the “how” and “why” these relationships between variables work.

ML algorithms broadly fall into one of two categories: Supervised Learning (SL) and Unsupervised Learning (UL).

SL models use training data with known outcomes and labels to make predictions when applied to new, similar data. By the use of a loss function, errors are identified for evaluation of the method and further improvement, wherein the minimisation of the loss function is the main goal [3]. Classification models categorise the input data based on the (discrete) given labels, creating boundaries to demarcate the different groups in the data. In this case, it would be whether a patient has undergone PCR or not. Regression models, on the other hand, use continuous data to find trends in the data, by estimating the relation between the dependent and independent variables. Regression will be used here to predict estimates of the RFS values.

UL is where a model finds undefined patterns or clusters in data, without any further user input, typically used in an exploratory fashion. This is done to deduce functions to explain hidden patterns from unlabelled data. For example, Principal Component Analysis (PCA) is used in dimensionality reduction, to find the most important features and remove the redundant ones, whereas clustering finds the unlabelled groups in the data. This project makes less use of UL and more of SL.

**3. METHODS**

**3.1 Dataset information**

The dataset that was used was based on the public dataset from The American College of Radiology Network, where a simplified dataset was generated from the I-SPY 2 TRIAL set. Each patient in the dataset had 10 clinical features (Age, ER, PgG, HER2, TrippleNegative Status, Chemotherapy Grade, Tumour Proliferation, Histology Type, Lymph node Status and Tumour Stage), and 107 image based features. These image-based features were extracted from the tumour region of MRI scans.

**3.2 Data Pre-processing**

Standard data pre-processing involves cleaning, preparing, and assessing the data for use. This is done by removing bad or missing data, identifying outliers, and sorting the data. In the dataset, PCR outcomes were given as 1 or 0, but missing data was represented by 999 or null values. Three different approaches could be used to combat this: the values could be replaced by the mean or median of the row, or just drop the value.

The dataset also needs to be normalised. Min-max, given by the equation , transforms the minimum values of features to 0, the maximum to 1, and the rest to decimals in between. It can be made more robust to outliers if the 5th and 95th percentiles are used. Z-normalisation takes , where each value is changed such that the overall mean, , is 0, and the standard deviation, , is 1 for the normalised values. This is done to minimise the effects that outliers have.

The data was split 80:20 for training and testing respectively.

**3.3 Feature Selection**

Not all features are generated equally. Some features may be irrelevant or redundant, leading to overly complicated models and a waste of computational budgets. Removing features with high correlation (over 0.95, using absolutes to account for negative values) is one way to remove features.

PCA is used to reduce the dimensions of the data, to circumvent the curse of dimensionality. Performance is negatively affected by high or low feature counts. It is a double-edged sword, however, since the removal of some features might be detrimental to the model as a whole, making the picture less clear and fundamentally less useful.

On the other hand, a feature selection approach involves removing noise from the data and using only relevant features in the model. Correlation was used for feature selection, which is a filter-based approach. It determines how linear the relationship between 2 variables is. Highly correlated features have a more linear dependence and therefore almost the same impact on the dependent variable, which means that one of these two features can be dropped, without much consequence.

**3.4 ML Algorithms**

Several models were considered in order to complete the project. Only one had to be picked for each task of PCR and RFS, with the former requiring a classification model and the latter a regression model. ML models can be used as both classifiers and regressors but require different hyperparameter tuning and differences in programming. Hyperparameter tuning was performed on all models, to find the combinations that gave the best accuracy.

*3.4.1 Support Vector Machines (SVM)*

SVM belongs to the SL category, combining statistical learning theory with generalisation [4]. The kernel trick makes SVM methods very useful. The way it works is to simulate the transformation of non-linearly separable data to higher dimensions by computing decision boundaries in terms of similarity measures for the higher dimensions. Since it avoids the costly computations, it is known as the kernel ‘trick’ [5].

The hyperparameters that have been tuned for the classifier (SVC) model are the kernel, which has been set to “poly” and since the kernel is polynomial, the coef0 hyperparameter (the independent term in the kernel function [13]) is set to 0.001. Gamma is set to “auto,” which is the default, and degree to 5. The regularisation term, , is set to 1, where the strength of regularisation decreases as this term increases in value. A squared L2 penalty is applied here [13].

The hyperparameters for the regressor (SVR) model have been tuned to use the “rbf” (Radial Basis Function) kernel, with and gamma set to “scale”,

*3.4.2 Artificial Neural Networks (ANN)*

ANNs are, unsurprisingly, neural networks (NNs). The NNs used were Multi-Layer Perceptron (MLP) NNs. The MLP is a fully connected feed forward network, with three layers: the input, output, and hidden layers. MLPs can solve non-linearly separable problems since they are designed to approximate non continuous functions. The models are trained by adjusting their weights and biases, to minimise error through the process of backpropagation [6].

NNs have a wide range of hyperparameters. The activation function was set to ReLU for all layers except the final layer, which was set to a sigmoid. The number of layers, which corresponds to the depth of the NN, was set to 5 dense and 3 dropout layers and the neurons per layer, corresponding to the width, was varied over the layers, with 1, 64, 128 and 256 being the possible number of neurons. Dropout was set to 10%. The same model was used for regression, but with ReLU for the output activation function.

*3.4.3 Decision Trees (DT) and Random Forest (RF)*

DTs are also an SL method. This algorithm uses a tree-like model to make and represent decisions. The decision rules inferred from the features of the given data are used to make a prediction [7].

The hyperparameters for the DT are chosen as follows, for classification and regression [8]:

chooses the function to assess the quality of the split. Set to “entropy” for Shannon Information Gain.

sets the maximum depth of the tree. This was set to 6.

is the minimum number of observations needed in each node to split it. This was set to 4 samples.

represents the minimum number of samples that need to be at a leaf node after it has been split. This was set to 3.

: provides the maximum number given to a tree when looking for the best split. Leaving this to the default gives , where is the number of features.

RF is an ensemble learning method which works by creating multiple decision trees during the training procedure, that mitigates the problem of overfitting in DTs by virtue of using many of them together [9].

For classification tasks, the class selected by most trees is chosen to be the overall output, and the mean of the values predicted by each tree is chosen to be the overall mean.

Since RF is an extension of DT, the hyperparameters will include the DT parameters. RFs also have some additional hyperparameters [9]:

limits the splitting of the nodes within a tree to restrict its growth. Setting this to none results in an unlimited number of leaf nodes.

represents the total number of trees in the forest, which was chosen to be 100.

is only required if is set to . It represents the number of samples to be taken from the original dataset to train each base estimator. 0.85 was used.

takes a Boolean value which is set to TRUE by default. In this case, is used when building trees. If this is set to False, the whole dataset is used for building every DT in the forest.

*3.4.4 Logistic and Linear Regression*

In statistics, logistic regression is a method which uses the given input variable to model probability of discrete outcomes. It is very commonly used for binary classification. . When using sklearn, different solvers can be set, and the optimal model can be found by comparing results. Different types of regularisation methods can be used to further improve the model [10]. In this project, the hyperparameters were set to use an L2 penalty and the “saga” solver.

Linear regression, on the other hand, is used for predicting quantitative values by forming linear relationships with either one or multiple independent features [11]. A standard linear regression model does not have any hyperparameters. There are variations on Linear Regression, such as LASSO (Least Absolute Shrinkage and Selection Operator), which was considered but ultimately not used.

*3.4.5 Extreme Gradient Boosting (XGBoost)*

XGBoost (Extreme Gradient Boosting) is an optimized and distributed implementation of Gradient Boosting framework. Boosting is an ensemble model that combines weak learning DTs. Each new tree is a fit on a modified version of the original data set. Gradient Boosting trains many models in an additive and sequential manner. The fundamental concept of this technique is to build the new base-learners to have a maximum correlation with the ensemble's overall negative gradient of the loss function. The trees are built in parallel, allowing a level-wise approach to be taking, meaning that the model scans over gradient values and assesses the quality of splits at each potential split in the training set using partial sums.

For the classifier, the only hyperparameters that were changed were the objective, set to “binary: logistic” and the random state, set to 42.

The regressor made more use of the hyperparameters [12]. The objective was set to “reg: squared error”. The same hyperparameter from RF, set to 100, with set to 7. The other hyperparameters of set to 0.4, to 0.1, seed to 10, and to 0.7. These extra hyperparameters are for the subsampling of columns and for the learning rate.

*3.4.6 K-Nearest Neighbours (KNN)*

KNN classifies new data depending on its proximity from neighbouring K points of the training data. Hence, if the majority of the nearest data belongs to a particular class, then this algorithm will predict that the new data also belongs to a particular class. [13]

KNN has been used for both classification and regression.

In sklearn, represents the number of neighbours for each query [14] The value of this hyperparameter was set to be 5 for both the classification and regression tasks.

**3.5 Evaluation Metrics**

The metrics used to evaluate the ML methods are Mean Absolute Error (MAE) for regression models, and Classification Accuracy for classifier models. K-fold Cross Validation was also used, where the standard folds was used. In addition, R2 score was used for regression models.

**4. RESULTS**

A multitude of models were tested, with 6 competing models for each task. In order to find the most accurate model, the ML algorithms were compared to each other.

Text

Description automatically generatedInitially, it appeared that the ANN was the best model to use for PCR, with around 90.3% accuracy, with the RF, DT and SVM at around 78.4% and 77.2% accurate. Eventually, the results in Figure 1 were obtained, with the ANN and XGBoost both getting 100% accuracy on the training data, and significantly worse on the K-fold validation accuracy. From these data, the best model was chosen to be SVC, with 75.7% accuracy in training and the K-fold validation accuracy.

Figure 1: The Classification and K-fold Cross Validation Accuracies for the Classifiers

For estimating RFS, which was a challenging task, the data provided in Figure 2 are not particularly promising. From the data, RF produced the best result, giving the lowest MAE. The R2 score was negative all across the board, but the RF model had the least negative value, which reinforced RF as the best choice.

Text

Description automatically generated

Figure 2: The MAE and R2 Score for the Regressors

A Receiver Operating Characteristic (ROC) curve was to be plotted but proved to be difficult to implement. This would have been an extra metric to assess the model performance.

**5. Discussion**

The choices made for data pre-processing were to use the robust scaler that can handle outliers better than Z-normalisation, replacing the mean with the median and the standard deviation with the interquartile range, in order to remove any influence that outliers might have. The null and missing values were replaced with the median and feature selection was chosen over PCA. This is due to the fact that PCA gave no clear improvement on the models in question and using Z-normalisation is not robust enough against outliers.

In order to check model fits, the training and validation accuracies were compared. From this comparison the ANN and XGBoost models were clearly overfitting. After simplifying the neural network model, it was inferior to the other models, and the other models were outperforming it. In the end, SVC was chosen for PCR, with the same accuracy for both criteria of 75.7%. Even though Logistic Regression had a higher accuracy, the small discrepancy in performance between the accuracies meant that there was less confidence in its ability to work properly.

Evidently, the values for RFS estimation are not reliably accurate. This could be down to a large array of issues. Since the R2 score was not a required accuracy test, it could have been disregarded. However, the values that were being produced were negative for all models, which could indicate that the dataset was possibly not suited for regression.

RF provided the best results, with the lowest MAE and least negative R2 score.

SVM models tend to be good at the given tasks (SVC and SVR for classification and regression respectively) and provide good accuracy for uninterpretable data. However, they are best used for smaller datasets with fewer features, which might explain why the model was not higher in accuracy.

RF is more suited to larger datasets with more features, and tends to provide more accurate results, but are slow to train. It is also less prone to overfitting.

A potential avenue of improvement for the MLP model could have been to look at ADAM optimisation, which might

**6. Conclusion**

The best model for the PCR prediction was the SVC, with a final classification accuracy of 75.7%, and the best model for RFS estimation was RF, which had a final MAE of 20.73. These are the models that will be submitted for the final performance evaluation. While it is the case that the RFS models may have underperformed, any further improvements suggested cannot be implemented due to time constraints.

# References

|  |  |  |
| --- | --- | --- |
| [1] | National Health Service, "Overview: Breast Cancer in Women," 28 October 2022. [Online]. Available: https://www.nhs.uk/conditions/breast-cancer/. [Accessed 4 November 2022]. | |
| [2] | | A. M. Menzies, R. N. Amaria, E. A. Rozeman, A. C. Huang, M. T. Tetzlaff, B. A. v. d. Wiel, S. Lo, A. A. Tarhini, E. M. Burton and T. E. Pennington, "Pathological response and survival with neoadjuvant therapy in melanoma: a pooled analysis from the International Neoadjuvant Melanoma Consortium (INMC)," *Nature Medicine,* vol. 27, pp. 301-309, 2021. |
| [3] | | J. A. M. Sidey-Gibbons and C. J. Sidey-Gibbons, "Machine Learning in Medicine: A Practical Introduction," *BMC Medical Research Methodology,* vol. 64, no. 19, 2019. |
| [4] | | C. Junli and J. Licheng, "Classification Mechanism of Support Vector," in *Proceedings of ICSP 2000*, 2000. |
| [5] | | S. Ghosh, A. Dasgupta and A. Swetapadma, "A Study on Support Vector Machine based Linear and Non-Linear Pattern Classification," in *International Conference on Intelligent Sustainable Systems 2019*, 2019. |
| [6] | | D. Kailita, "Analystics Vidhya," 30 March 2022. [Online]. Available: https://www.analyticsvidhya.com/blog/2022/03/an-overview-and-applications-of-artificial-neural-networks-ann/. [Accessed 12 December 2022]. |
| [7] | | L. Breiman, "RANDOM FORESTS," *University of California, Statistics Department,* January 2001. |
| [8] | | Scikit-learn Developers, "sklearn.tree.DecisionTreeClassifier," [Online]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.tree.DecisionTreeClassifier.html. [Accessed 5 December 2022]. |
| [9] | | Scikit-learn Developers, "sklearn.ensemble.RandomForestClassifier," [Online]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html. [Accessed 05 December 2022]. |
| [10] | | Scikit-learn Developers, "sklearn.linear\_model.LogisticRegression," [Online]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.linear\_model.LogisticRegression.html. [Accessed 8 December 2022]. |
| [11] | | Abhigyan, "Understanding Linear Regression," Medium, 19 July 2020. [Online]. Available: https://medium.com/analytics-vidhya/understanding-the-linear-regression-808c1f6941c0. [Accessed 9 December 2022]. |
| [12] | | A. Gupta, "XGBoost Hyperparameters Explained," 28 April 2021. [Online]. Available: https://amangupta16.medium.com/xgboost-hyperparameters-explained-bb6ce580501d. [Accessed 12 December 2022]. |
| [13] | | A. Martulandi, "Data Driven Investor," 22 October 2019. [Online]. Available: 14 https://medium.datadriveninvestor.com/k-nearest-neighbors-in-python-hyperparameters-tuning-716734bc557f. [Accessed 10 December 2022]. |
| [14] | | Scikit-learn Developers, "sklearn.neighbors.KNeighborsRegressor," [Online]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.neighbors.KNeighborsRegressor.html. [Accessed 10 December 2022]. |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Task and Weighting | Data Pre-processing (10%) | Feature Selection (20%) | ML Method Development (30%) | Method Evaluation (10%) | Report Writing (30%) |
| Charlie Davies |  |  |  |  |  |
| Davlesh Jodhun |  |  |  |  |  |
| Kshitij Tiwari |  |  |  |  |  |
| Lithicka Anandavel |  |  |  |  |  |
| Rajat Goyal |  |  |  |  |  |

1. 11% for near PCR, 14% for partial PCR. These values are outside the scope of the project. [↑](#footnote-ref-1)