AMSI VACATION RESEARCH SCHOLARSHIPS 2022–23

SET YOUR SIGHTS ON RESEARCH THIS SUMMER

Integrating Genomic and Clinical Data to Improve Ovarian Cancer Patient Outcomes

Jecinta Jaarola

Supervised by Dr. Nicola Armstrong

Curtin University

Vacation Research Scholarships are funded jointly by the Department of Education and the Australian Mathematical Sciences Institute.

AMSI AUSTRALIAN MATHEMATICAL SCIENCES INSTITUTE

Contents

Abstract

Integrating clinical information with microarray data has been shown to result in an improved performance of machine learning prediction of breast cancer prognosis, exhibiting a synergetic effect. For this project, a variety of integrated and unintegrated machine learning models were evaluated to identify whether this result can also be obtained for ovarian cancer samples. More specifically, these models have been constructed on clinical information and genomic data *separately* as per a late integration strategy, and then the outputs of each model have been combined. Model performance was then evaluated and compared to unintegrated models only. Although the results produced in this study are mostly inconclusive, this project serves as a preliminary insight into the integration of clinical information and genomic data for prediction of prognosis for patients with ovarian cancer. Thus, these results may provide directions for future research to investigate whether combining clinical information and genomic data results in a better prediction of prognosis for ovarian cancer patients.

Introduction

Ovarian cancer is the most lethal form of gynaecological malignancy, with almost 314,000 women diagnosed with ovarian cancer in 2020 (*Ovarian cancer statistics,* 2022). Concurrently, ovarian cancer is the 7th most common cancer, and the 8th most common cancer related cause of death for women, worldwide (Ferlay et al., 2014). The five-year survival rate for late-stage ovarian cancer is estimated to be lower than 41%, and despite the survival rate for all solid tumours improving dramatically within the last 50 years, the 5-year survival rate for ovarian cancer has not improved since 1980 (*Ovarian Cancer Stages,* 2023*)* (Vaughan et al., 2011).

Along with the high lethality, this type of cancer is also one of the most difficult to detect, with most diagnoses occurring in the late stages of the disease (Jayde & Boughton, 2014). This is due to the *clinical* symptoms of ovarian cancer being commonly misidentified as other diseases (*Early detection of ovarian cancer,* 2023). Apart from this, there are no proven methods that evaluate whether a patient is at high risk of ovarian cancer, especially as 77% of ovarian cancer occurrences are nonhereditary (Walsh et al., 2011).

The prognosis of a patient with ovarian cancer is generally determined by evaluating the type and stage of the cancer, along with the general health and age of the patient (*Ovarian Cancer,* 2023). However, this is only an estimate of the prognosis of a patient, and there are currently no tools in-use today that can predict the prognosis of an ovarian cancer patient using machine learning.

However, with the advent of microarrays, a wide range of molecular biomarkers have been proposed to potentially indicate prognosis in ovarian cancer patients. Furthermore, several studies have discovered that clinical information such as tumour size may provide some insight into the prognosis of a patient with ovarian cancer (Wu et al., 2022).

Thus, research of integrated clinical and genomic models to improve prediction of several outcomes has been of great interest recently. A study published in 2022 found that 37 out of 124 integrated machine learning models outperformed clinical decision making when determining which patients will undergo optimal cytoreduction for ovarian cancer (Cardillo et al., 2022). Another study published in 2012 found that integrating clinical and gene expression data resulted in a synergetic effect on predicting breast cancer outcome (van Vliet et al., 2012). More recently, a study published in 2022 developed a range of machine learning models that used patient-reported outcome data to predict six-month prognosis for ovarian cancer patients, with an accuracy of 79% (Sidey-Gibbons et al., 2022).

Therefore, the aim of this project is to integrate genomic data and clinical information, to evaluate whether there is an improved performance in predicting the prognosis of patients with ovarian cancer. The results of this research have identified some future directions for research in this topic, which may ultimately lead to an analytical tool that decides whether a patient with ovarian cancer would have a good long-term survival, or a poor long term survival, using their clinical information and gene expression data.

Statement of Authorship

The project idea was suggested by Dr Nicola Armstrong. All other work was completed by Jecinta Jaarola, with supervision and guidance by Dr Armstrong.

Background

Data Used

The data used in this project was obtained from The Cancer Genome Atlas, which is a cancer genomics program that began in 2006 and has characterized over 20,000 primary cancers to date (*TCGA,* 2023). To import this data into R, I used the 'curatedOvarianData' package, which is a Bioconductor package that contains a wide range of gene expression data on patients with ovarian cancer, including TCGA data (Ganzfried et al., 2013). In total, there were 196 cases with complete clinical information and microarray data.

Microarrays

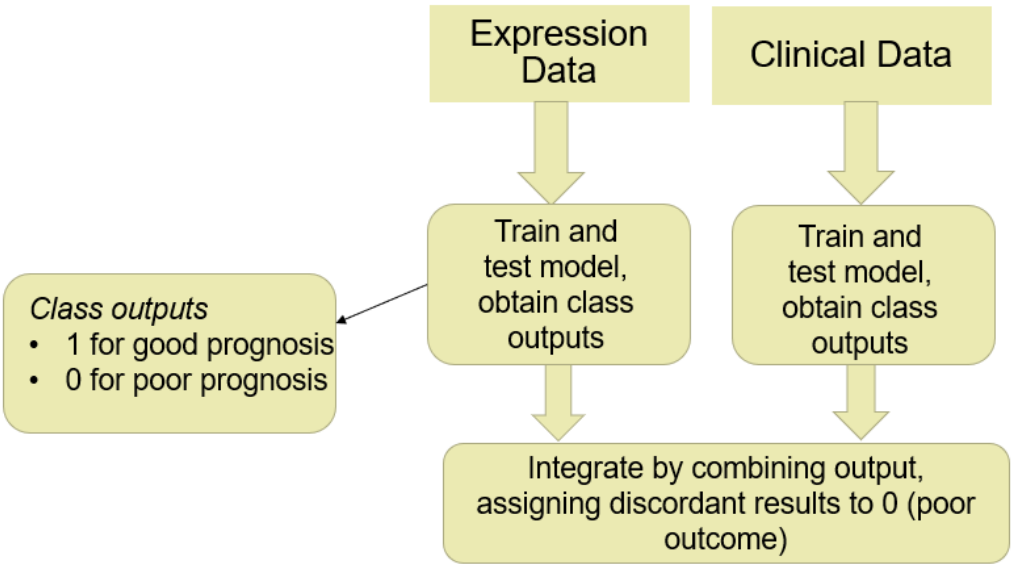
Microarrays are a recent development in the field of genomics, with the first concept being introduced in 1983 by Tse Wen Chang (Chang, 1983). These analytical tools have enabled clinicians to uncover biomarkers that may indicate the presence of a particular disease or help to evaluate an overall patient prognosis. More specifically, DNA microarrays are a set of human gene sequences arranged in a grid, in which the expression level of each gene is determined by measuring the interaction between the DNA molecules on the array, and the RNA molecules from the sample (Embl-Ebi, 2023). This is done by passing a laser through the microarrays and measuring the fluorescence of each DNA-RNA interaction (Kaliyappan et al., 2012). The microarrays used in this study are Affymetrix GeneChip Human Genome U133A microarrays, which contain over 14,500 genes (*Genechip™,* 2023).

Methods of Integration

There are three methods of integration clinical information and genomic data. The first method is early integration, where the datasets are concatenated together then analysed as a whole. Next is intermediate integration, in which the datasets are not concatenated, but rather the datasets are jointly analysed. Finally, late integration involves the analysis of each dataset separately, then the results of the analysis are combined (Chen et al., 2022).

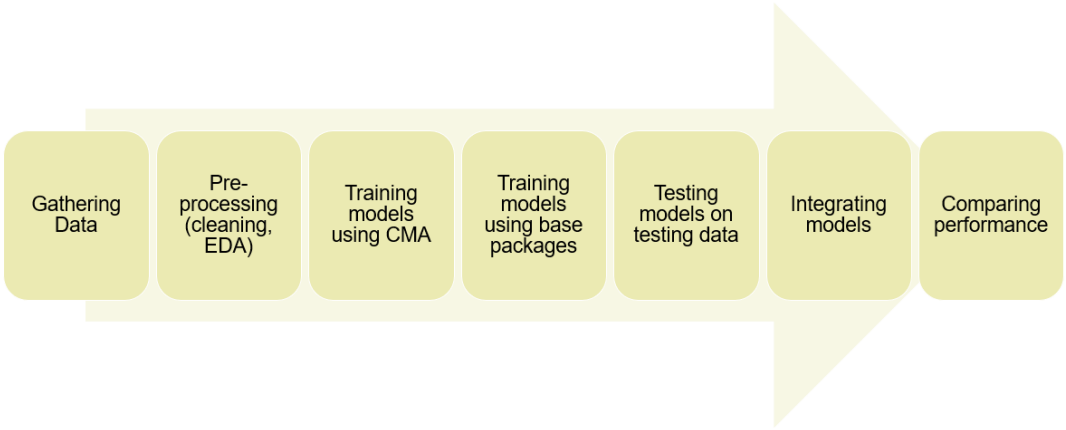
(Figure 0 – Integration Diagram) (Adossa et al., 2021)

According to van Vliet et al., the method of late integration seemed to yield the best results in their breast cancer study. For this project, late integration was undertaken by running machine learning classifiers on each dataset *separately*, then combining the class outputs for each classification model, using an AND/OR approach. More specifically, using an AND approach means that discordant results are assigned to the positive output (1), whereas using an OR approach means that discordant results are assigned to the negative output (0).



Methodology

There were seven phases of this project, as shown below.



Phase 1 – Obtaining the Data

As stated above, the data used in this project was sourced from The Cancer Genome Atlas, using the package 'curatedOvarianData' in R. This package allowed the data to be imported into R using several easy-to-use functions.

Phase 2 – Pre-processing & Exploratory Data Analysis

After importing the data, unnecessary or highly correlated variables were removed, resulting in a filtered dataset of clinical and genomic data. Next, the prognosis was assigned to each patient, those with a vital status of 'living' were assigned to the positive class (1), and all other cases assigned to the negative class (0). Before selecting this variable for prognosis, a variety of other variables were tried, such as assigning the positive class to patients with 'days to death' exceeding 3 years. However, the former classification specification resulted in the best performance for the classification models. A sample of 60 patients was selected from the dataset, to be used as the independent validation set for all models. The rest of the patients were designated to the training set. In this phase, exploratory data analysis was also conducted to discover any underlying patterns in the data. Firstly, a boxplot of the microarray dataset was generated, which identified an unusual sample as shown below.

(Figure 1 – Boxplot)

However, upon inspecting the clinical information for this case, it was found that this case didn't seem to be unusual, thus it was kept in the analysis.

A correlation matrix was also generated for this dataset, to see whether there were any correlated variables. Interestingly, it was found that 'days to death' and 'days to tumor recurrence' had a positive correlation of 0.71. This a high correlation, and means that an increase in the value of one variable results in an increase in the value of the other. Therefore, the 'days to death' variable was removed to reduce multicollinearity.

(Figure 2 – Correlation Matrix)

Phase 3 – Identifying Classifiers with the Best Performance

To identify the classifiers with the best performance, the CMA package was used. The CMA package is a 'wrapper' package that calls a variety of base packages, for synthesis of microarray-based classification (Slawski et al., 2022).

To begin, the classification models were trained on the genomic data first, using 5-fold cross validation. Fortunately, the CMA package was a very gentle introduction to machine learning, with variable selection, hyperparameter tuning and classification in one function. Thus, a classification model was constructed using each classifier in the CMA package, with a boxplot of the misclassification rates for the microarray data below.

(Figure 3 – Misclassification of Microarray Classifiers, excluding SVM)

As can be observed above, the random forest, elastic net and penalised logistic regression classifiers seemed to have the lowest misclassification. However, the support vector machine classifier was not able to be plotted due to a fault in the package.

(Figure 4 – Misclassification of Clinical Information Classifiers)

Shown above, the linear discriminant analysis and penalised logistic regression classifiers seemed to have the lowest misclassification for the clinical information. The neural network, lasso and elastic net classifiers also seemed to have a lower misclassification rate.

Phase 4 & 5– Constructing Classification Models

After observing the performance of each classifier in the previous phase, the top few classifiers with the best performance were selected, for the genomic data and the clinical information separately. During this process, the base packages were used to construct the classifier models, instead of using the CMA wrapper package. This process was undertaken as it means that the model objects can be refit onto the entire training set. These model objects allow the class outputs of each model to be compared, which is very important for late integration.

The performance of each model was evaluated using the independent validation dataset, which consists of 60 individuals, with 43 belonging to the negative class (poor long term prognosis) and 17 belonging to the positive class (good long term prognosis).
This phase was split into two parts, each focusing on constructing models with a single data type (genomic or clinical).

Phase 6 – Integrating the Models

In this phase, the class outputs of each classifier model were combined. More specifically, the class outputs of every microarray classification model were paired with the class outputs of every clinical information classification model. Cases with discordant classifications (i.e., 1 and 0, 0 and 1) were assigned to the poor long-term prognosis outcome. The resulting output was then compared to the **true** class of each case within the testing set, to evaluate the performance of the integrated model.

Phase 7 – Analysis of Results

This phase involved the creation of confusion matrices, ROC curves and several other performance metrics, for each of the integrated and unintegrated models. Using this information, it was possible to observe whether there was any synergetic effect arising from the integration of clinical information and genomic data.

Performance Metrics

Balanced Accuracy

Accuracy is simply a measure of the number of predictions that were correct. For example, an accuracy of 80% means that the model predicted the classes of 80% of all cases correctly (*Classification,* 2022). However, this performance metric is unsuitable for unbalanced datasets, where one class appears much more than the other class. In this case, around 33% of the cases in the complete dataset have a good long-term prognosis, with 35% in the training set and 28% in the validation set. Therefore, it is suitable to use *balanced accuracy*, which is the sum of the sensitivity and specificity of the model, divided by two (Zach, 2021). This performance metric provides a more correct evaluation of each model's performance.

Sensitivity and Specificity

Sensitivity is a measure of a model's ability to predict the class of true positives correctly (*Sensitivity,* 2022). For example, a highly sensitive model would predict few false negatives, which means that there are less cases of positives that are missed. In this scenario, a highly sensitive model can more easily discern that a patient would have a good long-term prognosis. Conversely, *specificity* is a measure of a model's ability to predict the class of true negatives correctly. In this case, a highly specific model can more easily classify a patient with poor long-term prognosis. For both performance metrics, a higher sensitivity and a higher specificity generally points towards a better-performing model. On the other hand, a model with a very high sensitivity and a very low specificity may be assigning the positive class to all cases, especially for an unbalanced dataset. Thus, a model with a high sensitivity **and** a high specificity is favorable. Yet, this is often unachievable as these performance measures are inversely related. Thus, it is important to achieve a balance between both performance metrics.

Area under the ROC Curve (AUC)

A very important performance metric for machine learning models is the *receiver operating characteristic (ROC) curve*. This is essentially a plot of the model's sensitivity versus the specificity, in which a model with a true positive rate of 1, and a false positive rate of 0, is perfect (*Classification,* 2022). The area under the ROC curve is known as the *AUC*, which is a general indicator of how well the model can distinguish between classes. A higher AUC means a better-performing model.

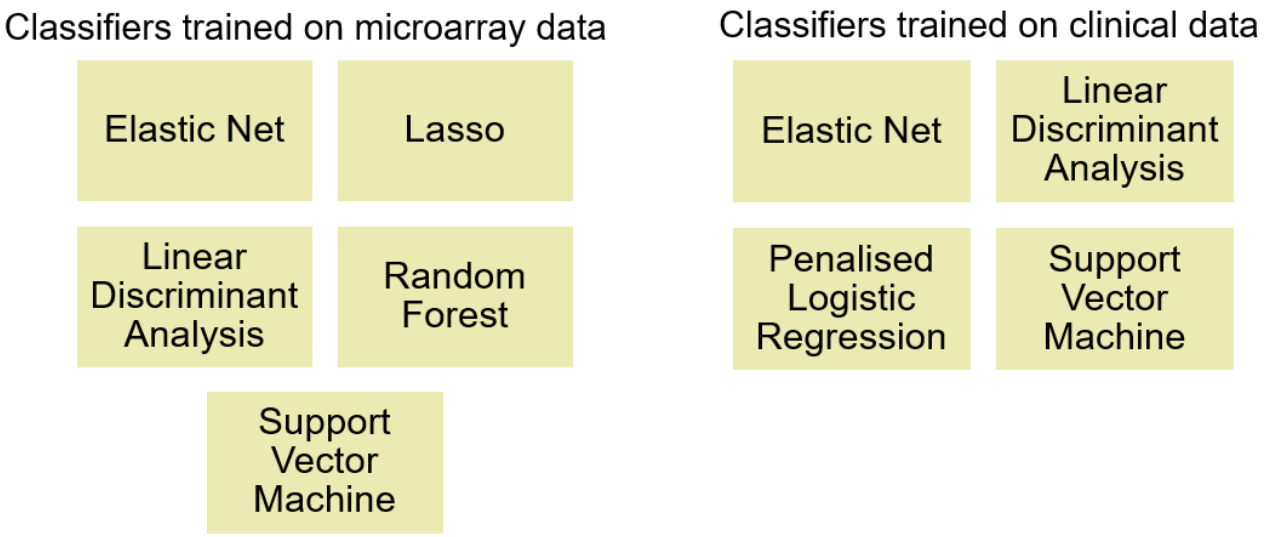
(Figure 4.1 – ROC Curve)

McNemar's Test

Another performance metric is the *McNemar's test*, which is a hypothesis test that evaluates whether the model is misclassifying one class more than another (Brownlee, 2018). For this test, a p-value below 0.05 indicates that there is strong evidence in support of the null hypothesis, which states that one class is being misclassified more than another class. A p-value above 0.05 means that the null hypothesis fails to be rejected, which conveys that the model is not misclassifying one class more than another. In this scenario,

a p-value above 0.05 is favourable, as it means that the model is not misclassifying the good long-term prognosis more than the poor long-term prognosis, despite an unbalanced dataset.

Results



Microarray Data

Random Forest

The first classifier that was selected for the microarray data was the random forest classifier, using the randomForest package in R (Liaw & Wiener, 2002). The random forest classifier consists of a collection of decision trees, where each decision tree aims to split the observations in a way that maximizes the differences between the resulting classes and minimizes the differences between the members of each class (Yiu, 2021). Thus, each decision tree provides a class prediction for a specific observation, and the class prediction with the most 'votes' out of all the decision trees, is the class output for the random forest classifier.

Before constructing a model, it was imperative to undertake hyperparameter tuning for this classifier. Thus, using the 'tuneRF' function in randomForest, it was found that the number of predictors to randomly sample at each split ('mtry') was 228, as this resulted in the smallest out-of-bag error.

(Figure 5 – Random Forest tuning)

To continue, a random forest model was constructed on the training data using the 'randomForest' function, with 'mtry' set to 228, and the number of trees set to 500, which is the default. The resulting out-of-bag error rate was 36.76%. The class error for predicting 0 was estimated to be 9%, whilst the class error for predicting 1 was estimated to be 87.5%. However, this performance metric is flawed, as the number of cases belonging to class 0 overwhelmingly outweighs the number of classes belonging to class 1 – thus, this classifier could be 'lazy', and simply predict all classes to be 0. This explains the very high error rate of predictions for class 1.

This model was then tested on the validation data, using the 'predict' function in base R. Simply observing the number of correct predictions versus incorrect predictions, this model predicted the classes of 44 out of 60 cases correctly – however, there are 43 cases with a true class of 0, meaning that this does not provide an appropriate indication of the performance of the model. Using a confusion matrix, it was found that the accuracy of this model was 73.33%, with a 95% confidence interval of (0.6034, 0.8393) (Kuhn, 2022). Yet, the balanced accuracy, which accounts for the imbalanced dataset, was only 60%. This model had a sensitivity of 29.41%, meaning that it predicts 29% of patients to have a good long-term prognosis. On the other hand, the specificity was 90.70%, meaning that it predicts 91% of patients to have a poor long-term prognosis. Furthermore, the McNemar's test resulted in a p-value of 0.08012, which means that the model is not misclassifying one class more than another. Finally, the AUC of this model was 0.660, which indicates a fair performance.

(Figure 6 – Random Forest Performance)

Elastic Net

After constructing a model using the random forest classifier, the elastic net classifier was applied, using the 'glmnet' package (Friedman et al., 2010). The elastic net classifier is essentially a combination of lasso and ridge regression, in which it uses penalties from both techniques to regularize regression models. Lasso regression (Least Absolute Shrinkage and Selection Operator) is a technique that adds the absolute magnitude of

coefficients as the penalty term for the loss function, whereas ridge regression adds the *squared* magnitude of coefficients as the penalty term for the loss function (Xu, 2021). The loss function is a function that maps the values of one or more variables onto a real number that represents the 'cost' associated with the event. In other words, it calculates the distance between the expected output of an algorithm, and its current output (Pere, 2020). Thus, elastic net regression model uses both penalties when selecting the constraint variables.

(Figure 7 – Elastic Net Diagram) (CFI Team, 2022)

The elastic net model requires two tuning parameters, the first being lambda to account for complexity, and the second being alpha, to set the compromise between lasso and ridge regression (Friedman et al., 2010). By default, alpha is set to 0.5. To obtain the lambda parameter, 5-fold cross validation was used, and the optimal lambda was found to be 0.22. The performance of this model was then evaluated using the validation data.

From the confusion matrix, the accuracy of this model was calculated to be 66.7%, with a 95% confidence interval of (0.5331, 0.7831). The sensitivity was 11.77%, with a specificity of 88.37%. The balanced accuracy was 50.07%, with a McNemar's test of 0.04, which means that the model misclassifies one class more than another. The AUC of this model was 0.501, which means that the model performs no better than random.

(Figure 8 – Elastic Net Performance)

Lasso

The 'glmnet' package was used once again to construct the lasso classification model, with alpha set to 1, to denote lasso classification. The lasso classifier, as discussed above, regularizes the regression coefficients by shrinking them, sometimes even reducing the coefficients to zero. In other words, the absolute magnitude of less important features may be shrunk to zero, which is optimal for feature selection.

Using the same process for elastic net, the accuracy of the lasso model was found to be 68.33%, with a 95% confidence interval of (0.5504, 0.7972). The sensitivity of this model

was 11.77%, with a specificity of 90.7%. Following, the balanced accuracy was 51.23%, with a McNemar's test p-value of 0.022, which suggests that the model was misclassifying one class more than the other. The AUC of this model was slightly better than the elastic net model, at 0.528, yet the performance is still no better than random.

(Figure 9 – Lasso Performance)

Support Vector Machine

After the lasso model, the support vector machine classifier model was constructed using the 'e1071' package in R (Meyer et al., 2023). The support vector machine algorithm aims to classify data points by locating a hyperplane in an N-dimensional space, where N is the number of features. The hyperplane is chosen by finding the maximum distance between observations of both classes, where data points on opposite sides of the hyperplane are assigned to different classes. Following, the observations that are adjacent to the hyperplane (i.e., the support vectors) directly affect the orientation and position of the hyperplane (Gandhi, 2018).

Hyperparameter tuning was conducted using the 'best.svm' function, which produced an optimised model which was then tested on the validation data. The accuracy of the model was found to be 75%, with a 95% confidence interval of (0.6214, 0.8528), w+hich is the best accuracy achieved so far. The sensitivity was found to be 17.65%, with a specificity of 97.67%. The balanced accuracy was unfortunately low, at 57.77%. The McNemar's test p-value was also very low, at 0.0019, which is strong evidence that the model is misclassifying one class more than the other. The AUC of this model was the highest of all models at 0.75, which suggests a good performance.

(Figure 10 – Support Vector Machine Performance)

Linear Discriminant Analysis

Using the 'MASS' package, the linear discriminant analysis model was constructed (Venables & Ripley, 2002). Linear discriminant analysis is a generalised form of Fisher's Linear Discriminant, which aims to maximise between-class scatter and minimise within-class scatter by assigning each data point onto a line. Essentially, linear discriminant

analysis uses a similar technique for multiple classes, by reducing *D* dimensional feature space to a *D'* dimensional feature space (where *D* is the number of features) to achieve the objective of minimising variability within classes and maximising variability between classes (Dash, 2022).

Linear discriminant analysis does not require hyperparameter tuning as it is a closed-form solution (*1.2. Linear and quadratic discriminant analysis*, 2023). Therefore, this model was created simply by using the 'lda' function and the 'predict' function. The resulting accuracy was 71.67%, with a 95% confidence interval of (0.5856, 0.8255). The sensitivity was 47.06%, with a specificity of 81.40%. The balanced accuracy was 64.23%. The McNemar's test p-value for this model was 1, therefore the model does not misclassify one class more than the other. The AUC of this model was 0.648, which is a fair performance.

(Figure 11 – Linear Discriminant Analysis Performance)

Clinical Data

Linear Discriminant Analysis

To begin, the 'MASS' package was used to create a linear discriminant analysis classification model on the clinical information, which was then tested on the testing data. As mentioned previously, no hyperparameter tuning was required for this classifier. The accuracy of this model was found to be 66.67%, with a 95% confidence interval of (0.5331, 0.7831). The sensitivity was found to be 47.06%, with a specificity of 74.42%. The balanced accuracy was determined to be 60.74%, with McNemar's test resulting in a p-value of 0.8231. The AUC of this model was 0.601, which is slightly better than random.

(Figure 11.1 – Linear Discriminant Analysis Performance – Clinical Data)

Elastic Net

Next, the elastic net classifier was modelled on the clinical information, with the performance evaluated on the testing data. The hyperparameter tuning produced a lambda of 0.0141, with an alpha of 0.5. The accuracy of this model was 66.67%, with a 95% confidence interval of (0.5331, 0.7831). The sensitivity was 47.06%, with a specificity of 74.42% and a balanced accuracy of 60.74%. Once again, the McNemar's test resulted

in a p-value of 0.8231, and the AUC was calculated to be 0.601. The statistics obtained in this model match the statistics obtained in the linear discriminant analysis model, which means that the models predicted the same classes for every observation.

(Figure 12 – Elastic Net Performance – Clinical Data)

Penalised Logistic Regression

To continue, the penalised logistic regression classifier was constructed and tested. Penalised logistic regression encompasses both lasso, ridge, and elastic net regression, by penalising the model for having too many variables, in which the coefficients of less important features are shrunk towards zero (Kassambara, 2018). For this case, the 'stepPlr' package was used to construct a classification model with the L2 penalty, which represents a ridge classification model (Park & Hastie, 2018). For this model, the accuracy was found to be 70%, with a 95% confidence interval of (0.5679, 0.8115). The sensitivity was 58.82%, with a specificity of 74.42%, and a balanced accuracy of 66.62%. Following, the McNemar's test resulted in a p-value of 0.4795, which conveys that this model did not misclassify one class more than the other. Finally, the AUC of this model was 0.648, which is a fair performance.

(Figure 13 – Penalised Logistic Regression Performance – Clinical Data)

Support Vector Machine

The last classifier for the clinical information was the support vector machine, using the 'e1071' package, in which the model was created using the 'best.svm' function, and evaluated using the 'predict' function. The accuracy of this model was found to be 66.67%, with a 95% confidence interval of (0.5331, 0.7831). Furthermore, the sensitivity was 23.53%, with a specificity of 83.72%. The balanced accuracy was calculated to be 53.63%, with a McNemar's test p-value of 0.2636. The AUC of this model was 0.549, which is a poor performance.

(Figure 14 – Support Vector Machine Performance – Clinical Data)

Integrating the Models

In this phase, thirteen different classifier combinations were constructed. To integrate the data, the class outputs of each of the models for the validation data (i.e. 60 individuals) were combined. For example, to combine the results from the elastic net classifier on microarray data, and the linear discriminant analysis classifier on clinical information, a data frame was created with two columns: the class outputs for the microarray data (essentially a vector of 0s and 1s) and the class outputs for the clinical data. If these columns had the same value for the class prediction (i.e. both 0 or both 1), then this value would be assigned as the combined predicted value. If the columns had different values for the class prediction (i.e. one prediction is 0, another is 1), then the poor outcome would be assigned as the combined predicted value (i.e. 0). The vector of combined predicted classes were then compared to the true classes of the 60 observations, to determine the performance of the new, integrated classification model.

Random Forest & Penalised Logistic Regression

The first combination of classifiers investigated was the random forest classifier on the genomic data, and the penalised logistic regression classifier on the clinical information. The accuracy of this model was determined to be 75% with a 95% confidence interval of (0.6214, 0.8528), with a sensitivity of 23.43%, and a specificity of 95.35%. Furthermore, the balanced accuracy was 59.44%, with a McNemar's test p-value of 0.009823. The AUC of this model was 0.713, which is a fair performance.

(Figure 15 – Random Forest & Penalised Logistic Regression)

Random Forest & Elastic Net

Secondly, the random forest classifier on genomic data paired with the elastic net classifier on the clinical information resulted in an accuracy of 71.67%, with a 95% confidence interval of (0.6214, 0.8528). The sensitivity was 11.77%, with a specificity of 95.35% and a balanced accuracy of 59.44%. Following, the McNemar's test p-value was 0.003609, and the AUC was 0.616.

(Figure 16 – Random Forest & Elastic Net)

Random Forest & Linear Discriminant Analysis

Next, the random forest classifier on genomic data combined with the linear discriminant analysis classifier on the clinical information resulted in an accuracy of 71.67% with a 95% confidence interval of (0.5856, 0.8255). The sensitivity of this model was 11.77%, with a specificity of 95.35%. Finally, the balanced accuracy was 53.56%, with a McNemar's test p-value of 0.003609 and an AUC of 0.616.

(Figure 17 – Random Forest & Linear Discriminant Analysis)

Random Forest & Support Vector Machine

Finally, the random forest classifier on genomic data paired with the support vector machine on clinical information resulted in an accuracy of 71.67%, with a 95% confidence interval of (0.5856, 0.8255). The sensitivity of this model was 5.88%, with a specificity of 97.67%. The balanced accuracy was 51.78%, with a McNemar's test p-value of 0.000685 and an AUC of 0.612.

(Figure 18 – Random Forest & Support Vector Machine)

Lasso & Penalised Logistic Regression

After integrating the random forest classifier on genomic data, the lasso classifier on genomic data was next. The first integrated model consisted of the lasso classifier, with the penalised logistic regression classifier on the clinical information. This resulted in an accuracy of 70%, with a 95% confidence interval of (0.5679, 0.8115). The sensitivity of this model was 5.88%, with a specificity of 95.35%. The balanced accuracy of this model was 50.62%, with a McNemar's test p-value of 0.002183 and an AUC of 0.526.

(Figure 19 – Lasso & Penalised Logistic Regression)

Lasso & Elastic Net, Lasso & Linear Discriminant Analysis

The next model consisted of the lasso classifier on genomic data, with the elastic net classifier on the clinical information. The model constructed after this was the lasso classifier on genomic data paired with the linear discriminant analysis classifier on clinical information. Interestingly, the accuracy, balanced accuracy, sensitivity, specificity and McNemar's test p-value for all three of the lasso classifier models were the same,

indicating that each integrated model had the exact same classifier output. After analysing the code numerous times, there was no fault to be found. Thus, the lasso classifier on genomic data resulted in the same performance across three different integrated models.

(Figure 19 – Lasso & Penalised Logistic Regression)

Elastic Net & Penalised Logistic Regression

Following, integrated models using the elastic net classifier on the genomic data, were created. The first of these models was the elastic net classifier and the penalised logistic regression classifier on clinical information. This model had an accuracy of 70%, with a 95% confidence interval of (0.5679, 0.8115). The sensitivity of this model was 5.88%, with a specificity of 95.35%. Finally, the balanced accuracy was 50.62%, with a McNemar's test p-value of 0.002183 and an AUC of 0.526.

(Figure 20 – Elastic Net & Penalised Logistic Regression)

Elastic Net & Elastic Net, Elastic Net & Linear Discriminant Analysis

The next integrated model was the elastic net classifier on the genomic data, paired with the elastic net classifier on the clinical information. However, it was found that this model, and the model with the linear discriminant analysis classifier on clinical information, all resulted in the same performance. This is the same case as the lasso classifier on genomic data and means that the predictions of each integrated model were the same. There was no fault found in the code.

(Figure 20 – Elastic Net & Penalised Logistic Regression)

Support Vector Machine & Penalised Logistic Regression

Finally, integrated models using the support vector machine classifier on the genomic data were investigated. To begin, the support vector machine combined with the penalised logistic regression classifier on clinical information resulted in an accuracy of 73.33% (with a 95% CI of (0.6034, 0.8393)). The sensitivity of this model was 11.77%, with a specificity of 97.67%. The balanced accuracy of this model was 54.72%, with a McNemar's test p-value of 0.001154 and an AUC of 0.702.

(Figure 21 – Support Vector Machine & Penalised Logistic Regression)

Support Vector Machine & Elastic Net

Next, the support vector machine classifier on the genomic data, and the elastic net classifier on the clinical information, were combined. This resulted in an accuracy of 71.67%, with a 95% confidence interval of (0.5856, 0.8255). This model had a sensitivity of 5.88%, a specificity of 97.67%, a balanced accuracy of 51.78% and a McNemar's test p-value of 0.000685. The AUC of this model was 0.612.

(Figure 22 – Support Vector Machine & Elastic Net)

Support Vector Machine & Linear Discriminant Analysis

The final integrated model was the support vector machine classifier combined with the linear discriminant analysis classifier on the clinical information. This integrated model produced the same statistics as the support vector machine combined with the elastic net classifier on the clinical information, which means that the same classes were predicted for each of the sixty cases, for these two models.

(Figure 22 – Support Vector Machine & Elastic Net)

Discussion

(Figure 23 – Results Table)

Evaluation using Performance Metrics

Balanced Accuracy

To begin, the models with the highest balanced accuracy were, in fact, the unintegrated models, with penalised logistic regression on the clinical information having the highest balanced accuracy of 67%. This was followed by linear discriminant analysis on the genomic data and the linear discriminant analysis on the clinical information, with balanced accuracies of 64% and 61% respectively. The highest balanced accuracy amongst the integrated models was much lower, at only 59%, as exhibited by the random forest classifier on genomic data paired with the penalised logistic regression on the clinical

information. Besides this, the average balanced accuracy of all models was 55%, which is an unsatisfactory result as it means that most models were performing only slightly better than random. Yet, the top two models have a balanced accuracy of above 60%. Overall, based on these results, it can be concluded that the unintegrated models provided the highest balanced accuracy.

Sensitivity

The top three models with the highest sensitivity were all unintegrated, with the highest measure belonging to the penalised logistic regression on the clinical information, at 58.82%. In other words, for this model, 58.82% of cases that were deemed to have a good long-term prognosis, actually did have a good long-term prognosis. As for the integrated models, the random forest classifier on the genomic data paired with the penalised logistic regression on the clinical information produced a sensitivity of 23.43%, which is a poor result. Unfortunately, the average sensitivity for all models was 18.45%, which conveys that most of the models were very poor at predicting a good long-term prognosis for patients that actually had a good long-term prognosis.

Specificity

There were five models with a specificity of 97.67%, meaning that 97.67% of cases predicted to have poor long-term prognosis, actually did have poor long-term prognosis. Four out of these five models were integrated models, with all integrated models having a higher specificity than eight out of nine of the unintegrated models. This makes sense, as all discordant results for these models were assigned to zero, indicating a poor long-term outcome. Thus, the integrated models performed better than the unintegrated models in this performance metric.

Area under the ROC Curve (AUC)

The model with the highest AUC was the support vector machine on the genomic data, with an AUC of 0.75. This means that, when taking two cases belonging to separate classes, there's a 75% chance that this model would be able to segregate the two cases by class. Following, the random forest paired with the penalised logistic regression model yielded an AUC of 0.713, followed by the support vector machine paired with the penalised logistic regression, with an AUC of 0.7018. Out of the top five models with the highest

AUC, there were only two integrated models. Yet, the worst-performing models were the unintegrated elastic net classifiers, with both models possessing an AUC of only 0.501. The average AUC for the unintegrated models was 0.599, with the average AUC for the integrated models being very slightly different, at 0.588. Therefore, it can be concluded that the unintegrated and integrated models had similar performance for distinguishing between classes.

McNemar's Test

There was a wide range of p-values produced for the McNemar's test, with all integrated models resulting in p-values below 0.05. This is strong evidence in favour of these models misclassifying one class more than the other: in this case, the good long-term prognosis. However, five of the nine unintegrated models had a p-value above 0.05, indicating that there's not enough evidence that one class was misclassified more than another, which is a favourable outcome. Thus, it can be concluded that the unintegrated models outperformed the integrated models in this aspect.

Ethical Concerns – Sensitivity versus Specificity

Before concluding which model is 'best', it is important to evaluate whether a higher specificity or a higher sensitivity should be favoured. That is, would we rather predict a patient to have a poor long-term prognosis, whereas they have a good long-term prognosis? Or would we rather predict a patient to have a good long-term prognosis, whereas they have a poor long-term prognosis?

It can be deduced that the situation in which a patient is predicted to have a good long-term prognosis, whereas the patient truly has a poor long-term prognosis, is more crucial than the converse scenario. This is because the patient may not receive the treatment that they need, as they were predicted to have a good long-term prognosis. Therefore, this patient may pass away due to a disease that could've been treated, had the prediction of their prognosis been correct.

In the opposite scenario, it can be evaluated that there are less consequences if a patient is predicted to have a poor-long term prognosis, but in-fact has a good long-term

prognosis. This may mean that the patient may enter palliative care and receive cancer treatment that is not needed. Yet, the patient is more likely to make a recovery and ultimately live, in this case. In the opposite scenario, the likelihood of this is very low.

In summary, a model with a higher sensitivity is much more favourable than a model with a higher specificity, as this means that there are fewer false positives, and thus there are fewer cases with poor-long term prognosis that are predicted to have a good long-term prognosis, which means fewer preventable deaths. For this project, thirteen of the seventeen integrated models had a higher specificity than the unintegrated models. Therefore, the integrated models may be more appropriate for use in real-life scenarios in this case, as these models predict less false positives, and thus should result in less deaths.

Top-performing Models

To evaluate the 'best' model, it is imperative for this model to have a high balanced accuracy, a high sensitivity **and** specificity, a high AUC and a McNemar's test p-value above 0.05. Although there were no models with good performance, there were *some* models that performed better than all the other models.

For this project, it was found that the best performing model was the penalised logistic regression on the clinical information. This model resulted in the highest balanced accuracy, at 66.62%, meaning that this model was correct in predicting classes for 66.62% of all cases in the testing set. To continue, this model had the highest sensitivity of 58.82%, meaning that the model was able to classify around 59% of true positives correctly. Thus, given a case with a good long-term outcome, there is a 59% chance that the model could predict this outcome correctly. Conversely, the specificity of this model was 74.42%, which means that the model was able to classify around 74% of the true negatives correctly. In other words, given a case with a poor-long term outcome, there is a 74% chance that the model could predict this outcome correctly. This model had the 5th highest AUC at 0.648%, which conveys that there is a 65% chance that the model could differentiate between two cases with different long-term prognoses. Finally, the McNemar's

test p-value for this model was 0.4795, indicating that this model did not misclassify one class more than another.

A strong contender for the second 'best' performing model was the linear discriminant analysis on genomic data. This model possessed the second-highest balanced accuracy of 64%, which is a fair performance. Furthermore, the sensitivity was 47.06%, meaning that the model was able to classify around 47% of the true positives correctly. Conversely, the specificity of this model was 81.40%, which means that the model was able to classify around 81.40% of the true negatives correctly. This model had the 6th highest AUC, at 64.77%, which means that there is a 64.77% chance that the model can segregate two cases with different long-term prognoses. The McNemar's test p-value for this model was 1, indicating that this model did not misclassify one class more than another.

Observing the results from the integrated models alone, it was found that the random forest classifier on the genomic data in combination with the penalised logistic regression classifier on the clinical information, was the 'best' integrated model. This model had a balanced accuracy of 59.44%, meaning that it correctly classified 59% of all cases in the testing set. This model also had the highest sensitivity of all integrated models, at 23.43%, which indicates that this model was able to classify around 23% of true positives correctly. On the other hand, the specificity of this model was 95.35%, meaning that this model was able to classify around 95% of true negatives correctly. The AUC of this model was 0.713, which is the highest out of all integrated models, and conveys that there is a 71.3% chance that this model can differentiate between two cases that have different classes. Unfortunately, the McNemar's test p-value for this model was 0.009823, which suggests that this model misclassified one case significantly more than the other.

Problems Encountered

Unfortunately, there were several problems that were encountered throughout this project. To begin, some of the classifiers that were evaluated to have a good performance according to CMA, were not able to be used in this project, as the base packages that were intended to be used to construct the models did not have the necessary functions for creating a model object or obtaining the class outputs. Thus, classifiers such as neural

network exhibited a good performance for misclassification, yet the base packages for these classifiers did not appear to have the necessary tools to easily continue with them. In the future, this problem may be solved by using different programs such as Python, to construct models using these classifiers.

Another problem in this project was the collection of data to use, as there were a very small number of publicly available datasets that contained microarray data and clinical information for patients with ovarian cancer. The requirements of the data included many cases and a high number of features for the clinical information, which only the TCGA dataset could fulfill. Unfortunately, even this dataset only had 196 cases and nine useful clinical features. In contrast, other studies such as van Vliet et al., used 259 samples for training the models, 521 samples for independent validation and 54 clinical features. Therefore, this project may have produced more conclusive results with a larger, more informative dataset.

One problem that was discovered was that some of the integrated models appeared to produce the exact same statistics: for example, the lasso classifier on genomic data. Thus, the exact same combined classification outputs were produced for some models. However, upon analysing the code and checking the arguments to each function, it was found that there was no fault in the code. Yet, it is important to have awareness of this occurrence in future projects, as it may indicate a problem with the code.

One problem that was discovered later in the project was that the package used for penalised logistic regression utilized the ridge regression penalty instead, therefore all the ‘penalised logistic regression’ models were ridge classifier models. This is due to a misunderstanding that penalised logistic regression was a separate classifier, when it is in-fact an umbrella term which includes several types of logistic regression such as lasso, ridge, and elastic net regression. For future studies, more extensive background research will be conducted into the classifiers, and how they work.

Final Evaluation

Comparing the performance of both types of models, it can be concluded that the unintegrated models seemed to perform better than the integrated models in all aspects. However, even the best-performing unintegrated models did not have a great performance overall, and the average AUC for all models was 0.5922, which is a poor performance. Nevertheless, it is crucial to consider whether the **data** itself contains the necessary information to train a well-performing machine learning model. Unfortunately, very few clinical features were used to train the classifiers, and there were only 196 cases in total. Therefore, with such a limited amount of data, it is important to assume that these results may not be reliable and should instead be used as a preliminary insight into further research in this topic.

Future endeavours

There are many directions that should be explored in relation to this project. To begin, it may be pertinent to combine classifiers of the same type, for example, using elastic net classifiers on both the genomic and clinical data, then integrating the models. The performance of the integrated model could then be compared to unintegrated models that use elastic net on genomic data and clinical data *separately*. This would mean that the effect of integrating the datasets could be observed *directly*, instead of using different classifier combinations.

Besides this, it may be beneficial to compare the performance of the integrated models, using early, intermediate, or late integration of microarray and clinical information. In this project, only late integration was used. Thus, there is an opportunity to investigate whether early or intermediate integration may result in better performance than unintegrated models.

Finally, another future direction is to follow the same process of this project, using a larger dataset. This would mean that patterns in the data are more likely to be discovered by the machine learning models, as there are more features and samples. Therefore, using a larger dataset may support, disprove or discover new findings in the integration of clinical and genomic data for ovarian cancer patients.

Conclusion

The purpose of this project was to attempt to improve the prediction of prognosis for ovarian cancer patients, by integrating clinical information and genomic data. Although there have been other studies that have shown a synergetic effect when combining these types of data for prediction of prognosis for breast cancer, it seems that this effect was not observed in the results for this project, as the unintegrated models outperformed the integrated models in most aspects. However, this may be due to a lack of information in the datasets used in this project, and thus further studies should be undertaken using a larger dataset. Therefore, this project serves as a preliminary insight into the integration of clinical and genomic data to improve the prediction of prognosis for patients with ovarian cancer, and provides direction for future research in this topic.

Bibliography

- *1.2. Linear and quadratic discriminant analysis* (2023) *Scikit Learn*. scikit-learn developers. Available at: https://scikit-learn.org/stable/modules/lda_qda.html (Accessed: February 23, 2023).
- Adossa, N. *et al.* (2021) *Computational strategies for single-cell multi-omics integration - Fig 2*. Computational and Structural Biotechnology Journal. Available at: https://www.researchgate.net/figure/Schematic-illustration-of-the-early-intermediate-and-late-data-integration-strategies-in_fig1_351135400 (Accessed: February 23, 2023).
- Brownlee, J. (2018) *How to calculate McNemar's test to compare two machine learning classifiers*, *MachineLearningMastery.com*. Guiding Tech Media. Available at: https://machinelearningmastery.com/mcnemars-test-for-machine-learning/ (Accessed: February 22, 2023).
- Cardillo, N. *et al.* (2022) "Integrated clinical and genomic models to predict optimal cytoreduction in high-grade serous ovarian cancer," *Cancers*, 14(14), p. 3554. Available at: https://doi.org/10.3390/cancers14143554.
- CFI Team (2022) *Elastic Net Diagram*, *Corporate Finance Institute*. CFI Education Inc. Available at: https://corporatefinanceinstitute.com/resources/data-science/elastic-net/ (Accessed: February 22, 2023).
- Chang, T.-W. (1983) "Binding of cells to matrixes of distinct antibodies coated on solid surface," *Journal of Immunological Methods*, 65(1-2), pp. 217–223. Available at: https://doi.org/10.1016/0022-1759(83)90318-6.
- Chen, Y.-A., Allendes Osorio, R.S. and Mizuguchi, K. (2022) "Targetmine 2022: A new vision into drug target analysis," *Bioinformatics*, 38(18), pp. 4454–4456. Available at: https://doi.org/10.1093/bioinformatics/btac507.
- *Classification: Accuracy | machine learning | google developers* (2022) *Google*. Google. Available at: https://developers.google.com/machine-learning/crash-course/classification/accuracy (Accessed: February 22, 2023).
- *Classification: Roc curve and AUC | machine learning | google developers* (2022) *Google*. Google. Available at: https://developers.google.com/machine-learning/crash-course/classification/roc-and-auc (Accessed: February 23, 2023).
- Dash, S.K. (2022) *Linear discriminant analysis: What is linear discriminant analysis*, *Analytics Vidhya*. Analytics Vidhya. Available at: https://www.analyticsvidhya.com/blog/2021/08/a-brief-introduction-to-linear-discriminant-analysis/ (Accessed: February 23, 2023).
- *Early detection of ovarian cancer* (2023) *Cancer Council*. Cancer Council. Available at: https://www.cancer.org.au/cancer-information/causes-and-prevention/early-detection-and-screening/early-detection-of-ovarian-cancer (Accessed: February 20, 2023).
- Embl-Ebi (2023) *Microarrays*, *Microarrays | Functional genomics II*. EMBL. Available at: https://www.ebi.ac.uk/training/online/courses/functional-genomics-ii-common-technologies-and-data-analysis-methods/microarrays/ (Accessed: February 21, 2023).
- Ferlay, J. *et al.* (2014) "Cancer incidence and mortality worldwide: Sources, methods and major patterns in Globocan 2012," *International Journal of Cancer*, 136(5). Available at: https://doi.org/10.1002/ijc.29210.
- Friedman, J., Hastie, T. and Tibshirani, R. (2010) "Regularization paths for generalized linear models via coordinate descent," *Journal of Statistical Software*, 33(1). Available at: https://doi.org/10.18637/jss.v033.i01.
- Friedman, J., Hastie, T. and Tibshirani, R. (2010) "Regularization paths for generalized linear models via coordinate descent," *Journal of Statistical Software*, 33(1). Available at: https://doi.org/10.18637/jss.v033.i01.
- Gandhi, R. (2018) *Support Vector Machine - introduction to machine learning algorithms*, *Medium*. Towards Data Science. Available at: https://towardsdatascience.com/support-vector-machine-introduction-to-machine-learning-algorithms-934a444fca47 (Accessed: February 22, 2023).
- Ganzfried, B.F. *et al.* (2013) "CURATEDOVARIANDATA: Clinically annotated data for the ovarian cancer transcriptome," *Database*, 2013. Available at: https://doi.org/10.1093/database/bat013.
- *Genechip™ Human Genome U133a 2.0 Array* (2023) *ThermoFisher Scientific - US*. ThermoFisher. Available at: https://www.thermofisher.com/order/catalog/product/900471 (Accessed: February 21, 2023).
- Jayde, V. and Boughton, M. (2014) "The diagnostic journey of ovarian cancer: A review of the literature and suggestions for Practice," *Contemporary Nurse*, 41(1), pp. 5–17. Available at: https://doi.org/10.5172/conu.2012.41.1.5.
- Kaliyappan, K. *et al.* (2012) "Microarray and its applications," *Journal of Pharmacy And Bioallied Sciences*, 4(6), p. 310. Available at: https://doi.org/10.4103/0975-7406.100283.
- Kassambara, A. (2018) *Penalized logistic regression essentials in R: Ridge, Lasso and elastic net*, *STHDA*. STHDA. Available at: http://www.sthda.com/english/articles/36-classification-methods-essentials/149-penalized-logistic-regression-essentials-in-r-ridge-lasso-and-elastic-net/ (Accessed: February 23, 2023).
- Kuhn, M. (2022) *caret: Classification and Regression Training*, *CRAN*. Available at: https://CRAN.R-project.org/package=caret (Accessed: 2022).
- Liaw, A. and Wiener, M. (2002) *Classification and Regression by randomForest*, *The R Journal*. CRAN. Available at: https://CRAN.R-project.org/doc/Rnews/ (Accessed: February 22, 2023).
- Meyer, D. *et al.* (2023) "e1071: Misc Functions of the Department of Statistics, Probability Theory Group (Formerly: E1071), TU Wien."

- *Ovarian cancer stages, survival rate and prognosis* (2023) *Ovarian Cancer Research Alliance*. Ovarian Cancer Research Alliance Inc. Available at: https://ocrahope.org/get-the-facts/staging/ (Accessed: February 20, 2023).
- *Ovarian cancer statistics: World cancer research fund international* (2022) *WCRF International*. World Cancer Research Fund International. Available at: https://www.wcrf.org/cancer-trends/ovarian-cancer-statistics/ (Accessed: February 20, 2023).
- *Ovarian cancer: Causes, symptoms & treatments* (2023) *Cancer Council*. Cancer Council. Available at: https://www.cancer.org.au/cancer-information/types-of-cancer/ovarian-cancer (Accessed: February 21, 2023).
- Park, M.Y. and Hastie, T. (2018) "stepPlr: L2 Penalized Logistic Regression with Stepwise Variable Selection."
- Pere, C. (2020) *What are loss functions?*, *Medium*. Towards Data Science. Available at: https://towardsdatascience.com/what-is-loss-function-1e2605aeb904 (Accessed: February 22, 2023).
- *Sensitivity and specificity of machine learning* (2022) *Deepchecks*. Deepchecks AI. Available at: https://deepchecks.com/glossary/sensitivity-and-specificity-of-machine-learning/ (Accessed: February 26, 2023).
- Sidey-Gibbons, C.J. *et al.* (2022) "Predicting 180-day mortality for women with ovarian cancer using machine learning and patient-reported outcome data," *Scientific Reports*, 12(1). Available at: https://doi.org/10.1038/s41598-022-22614-1.
- Slawski, M., Boulesteix, A.-L. and Bernau, C. (2022) "Synthesis of microarray-based classification." Available at: https://doi.org/10.18129/B9.bioc.CMA.
- *The cancer genome atlas program* (2023) *National Cancer Institute*. U.S. Department of Health and Human Services. Available at: https://www.cancer.gov/about-nci/organization/ccg/research/structural-genomics/tcga (Accessed: February 21, 2023).
- van Vliet, M.H. *et al.* (2012) "Integration of clinical and gene expression data has a synergetic effect on predicting breast cancer outcome," *PLoS ONE*, 7(7). Available at: https://doi.org/10.1371/journal.pone.0040358.
- Vaughan, S. *et al.* (2011) "Rethinking ovarian cancer: Recommendations for improving outcomes," *Nature Reviews Cancer*, 11(10), pp. 719–725. Available at: https://doi.org/10.1038/nrc3144.
- Venables, W.N. and Ripley, B.D. (2002) "MASS," *Modern Applied Statistics with S* [Preprint].
- Walsh, T. *et al.* (2011) "Mutations in 12 genes for inherited ovarian, fallopian tube, and peritoneal carcinoma identified by massively parallel sequencing," *Proceedings of the National Academy of Sciences*, 108(44), pp. 18032–18037. Available at: https://doi.org/10.1073/pnas.1115052108.
- Wu, L. *et al.* (2022) "Tumor size is an independent prognostic factor for stage I ovarian clear cell carcinoma: A large retrospective cohort study of 1,000 patients," *Frontiers in Oncology*, 12. Available at: https://doi.org/10.3389/fonc.2022.862944.
- Xu, W. (2021) *What's the difference between linear regression, Lasso, Ridge, and ElasticNet?*, *Medium*. Towards Data Science. Available at: https://towardsdatascience.com/whats-the-difference-between-linear-regression-lasso-ridge-and-elasticnet-8f997c60cf29 (Accessed: February 22, 2023).
- Yiu, T. (2021) *Understanding random forest*, *Medium*. Towards Data Science. Available at: https://towardsdatascience.com/understanding-random-forest-58381e0602d2 (Accessed: February 22, 2023).
- Zach (2021) *What is balanced accuracy?* , *Statology*. Statology. Available at: https://www.statology.org/balanced-accuracy/ (Accessed: February 22, 2023).

Appendix

Figure 0 – Integration Diagram

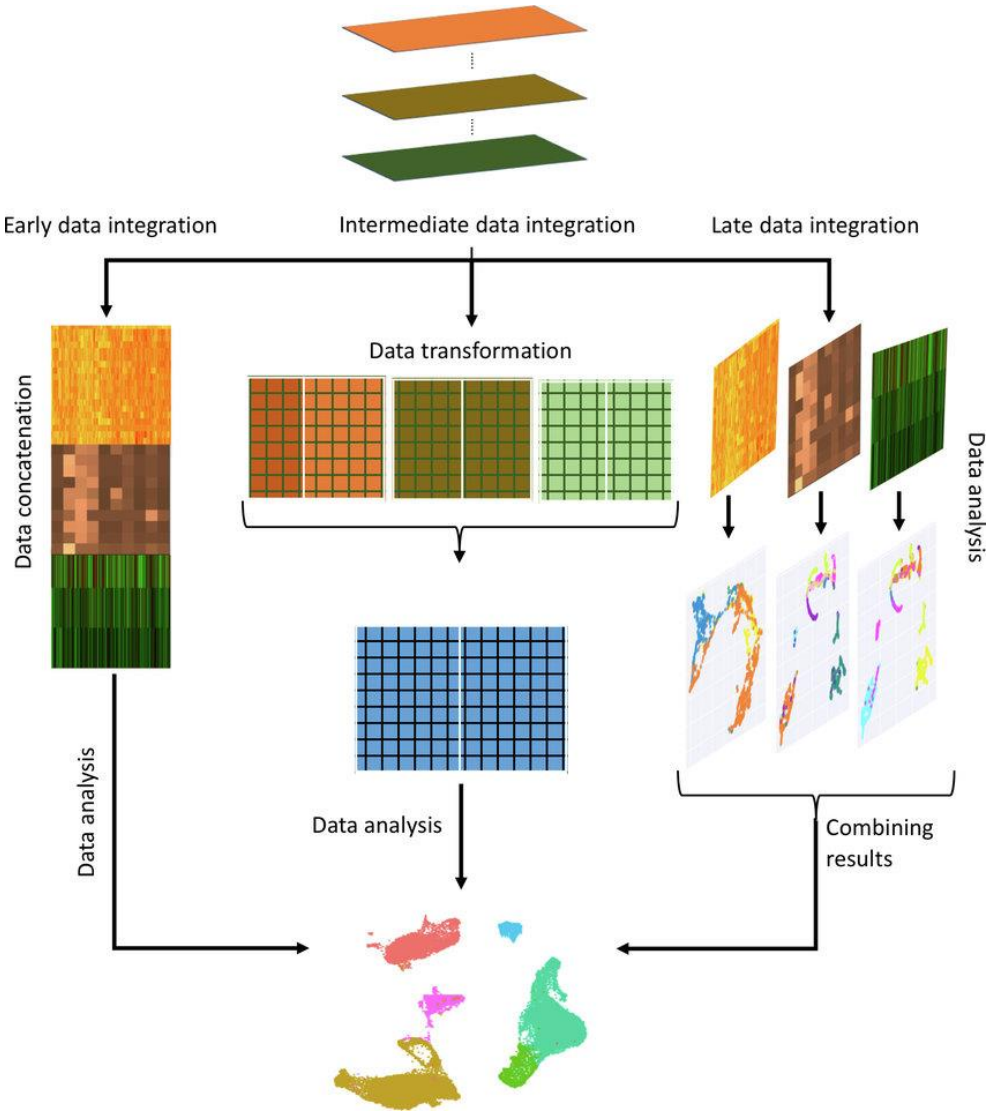


Figure 1 – Boxplot

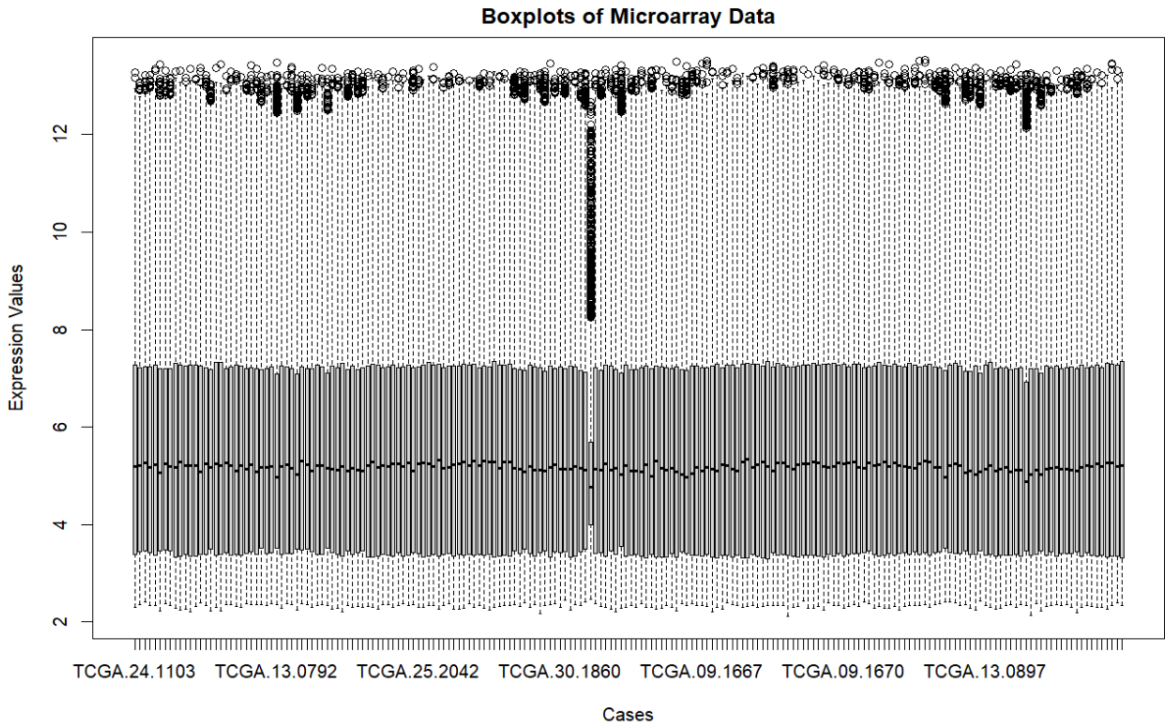


Figure 2 – Correlation Matrix

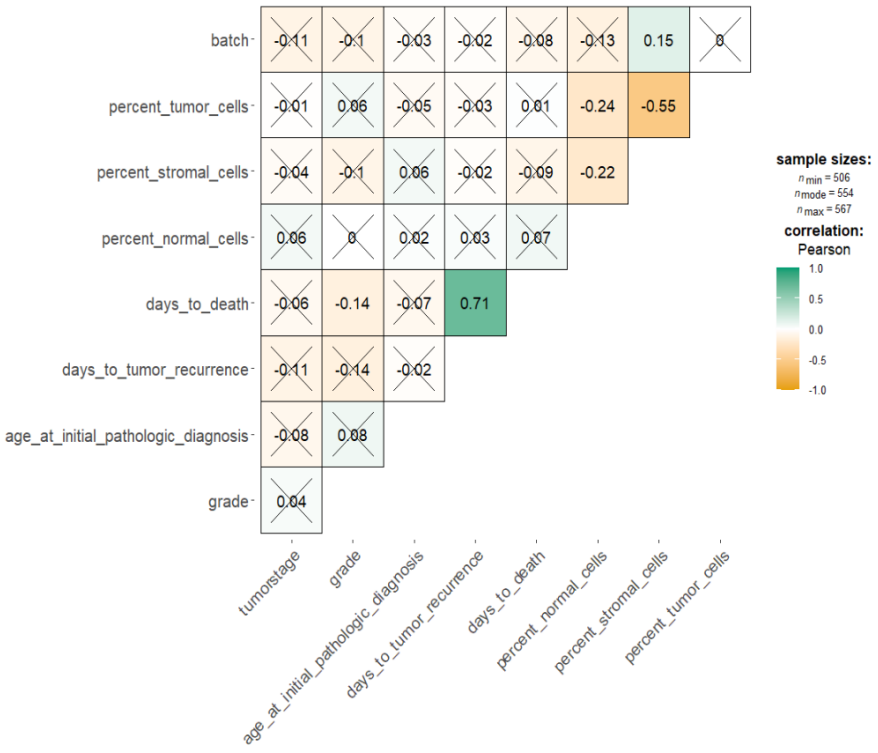


Figure 3 – Misclassification of Microarray Classifiers, excluding SVM

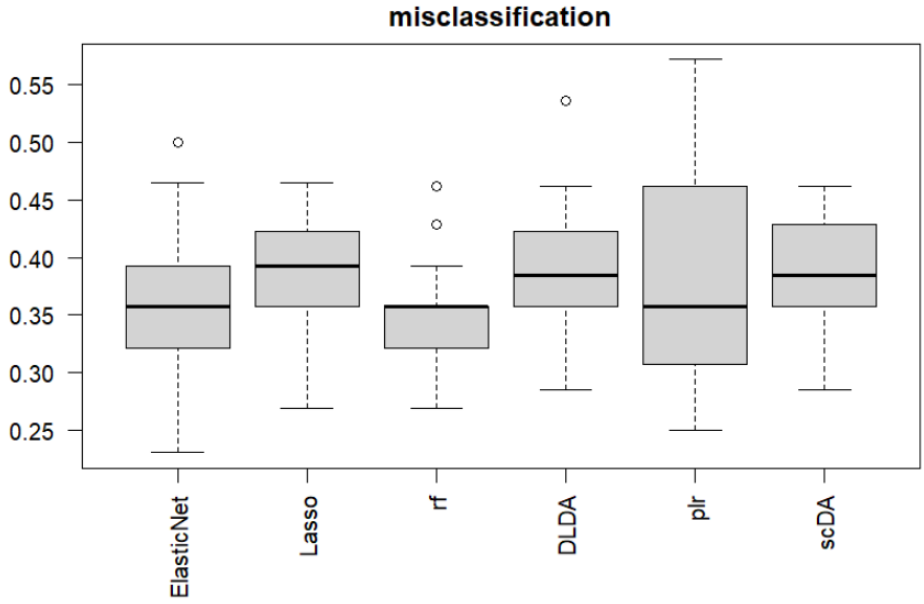


Figure 4 – Misclassification of Clinical Information Classifiers, excluding SVM

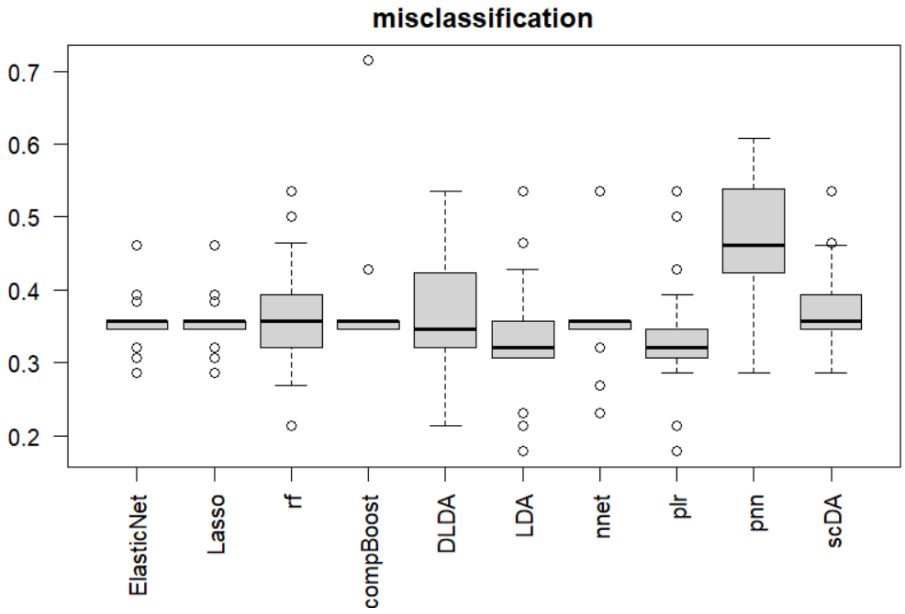


Figure 4.1 – ROC Curve

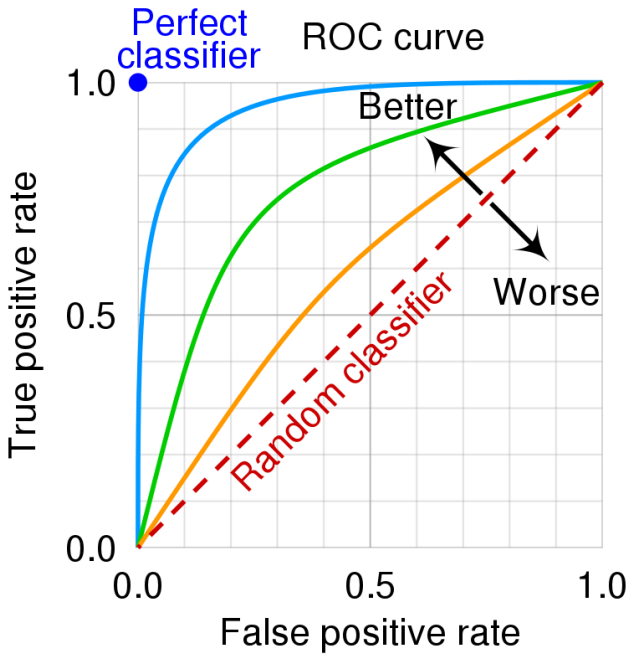


Figure 5 – Random Forest tuning

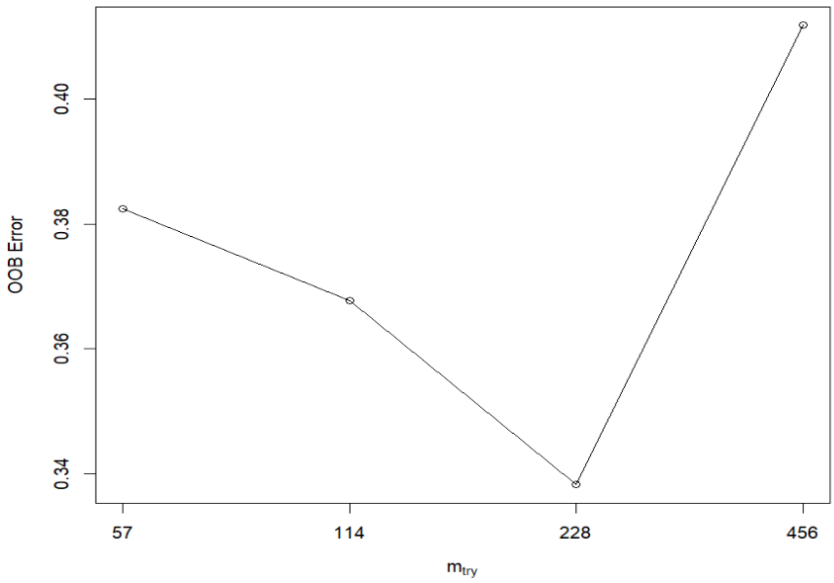


Figure 6 – Random Forest Performance

```
               Accuracy : 0.7333
                 95% CI : (0.6034, 0.8393)
    No Information Rate : 0.7167
    P-Value [Acc > NIR] : 0.45124

                  Kappa : 0.2344

 Mcnemar's Test P-Value : 0.08012

            Sensitivity : 0.29412
            Specificity : 0.90698
         Pos Pred Value : 0.55556
         Neg Pred Value : 0.76471
             Prevalence : 0.28333
         Detection Rate : 0.08333
   Detection Prevalence : 0.15000
      Balanced Accuracy : 0.60055
```

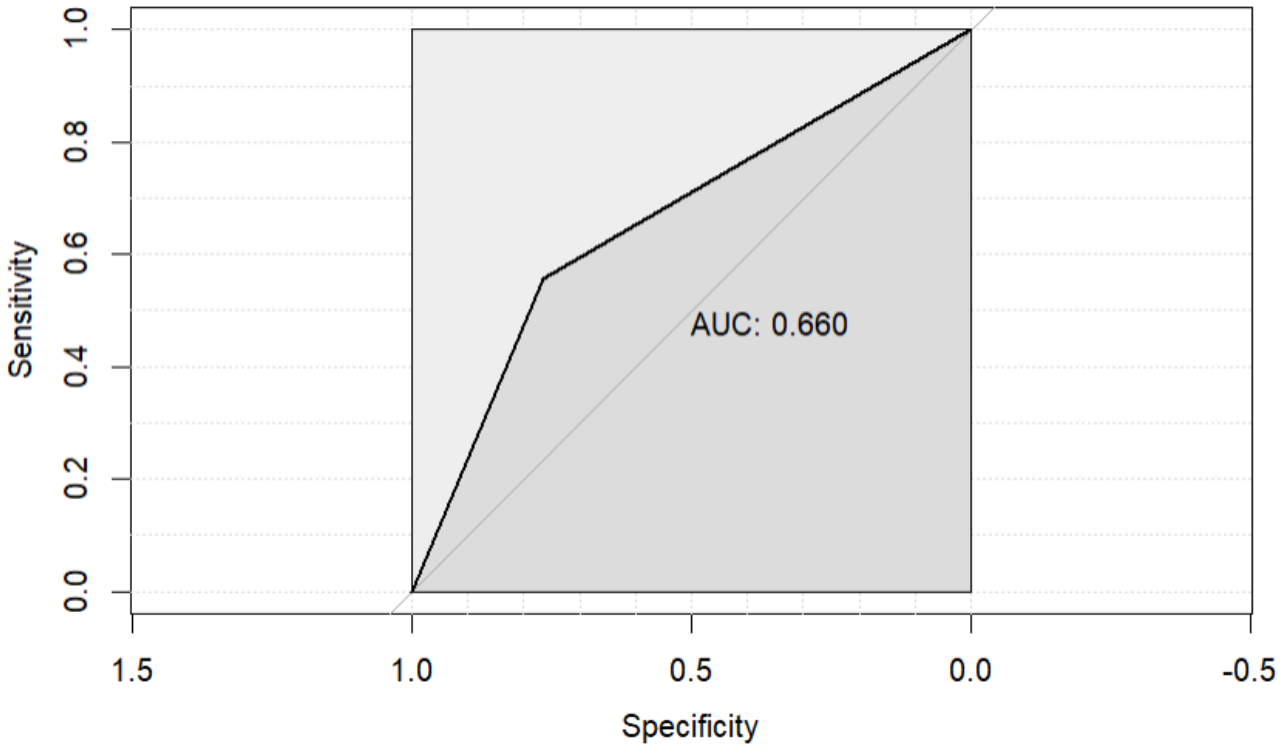


Figure 7 – Elastic Net Diagram

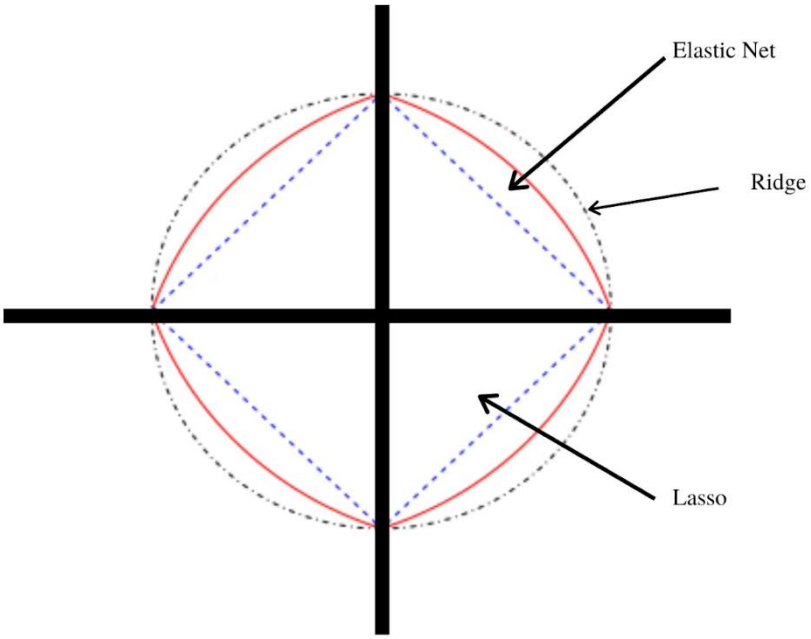


Figure 8 – Elastic Net Performance

```
                Accuracy : 0.6667
                  95% CI : (0.5331, 0.7831)
    No Information Rate : 0.7167
    P-Value [Acc > NIR] : 0.84214

                   Kappa : 0.0017

 Mcnemar's Test P-Value : 0.04417

            Sensitivity : 0.11765
            Specificity : 0.88372
         Pos Pred Value : 0.28571
         Neg Pred Value : 0.71698
             Prevalence : 0.28333
         Detection Rate : 0.03333
   Detection Prevalence : 0.11667
      Balanced Accuracy : 0.50068
```

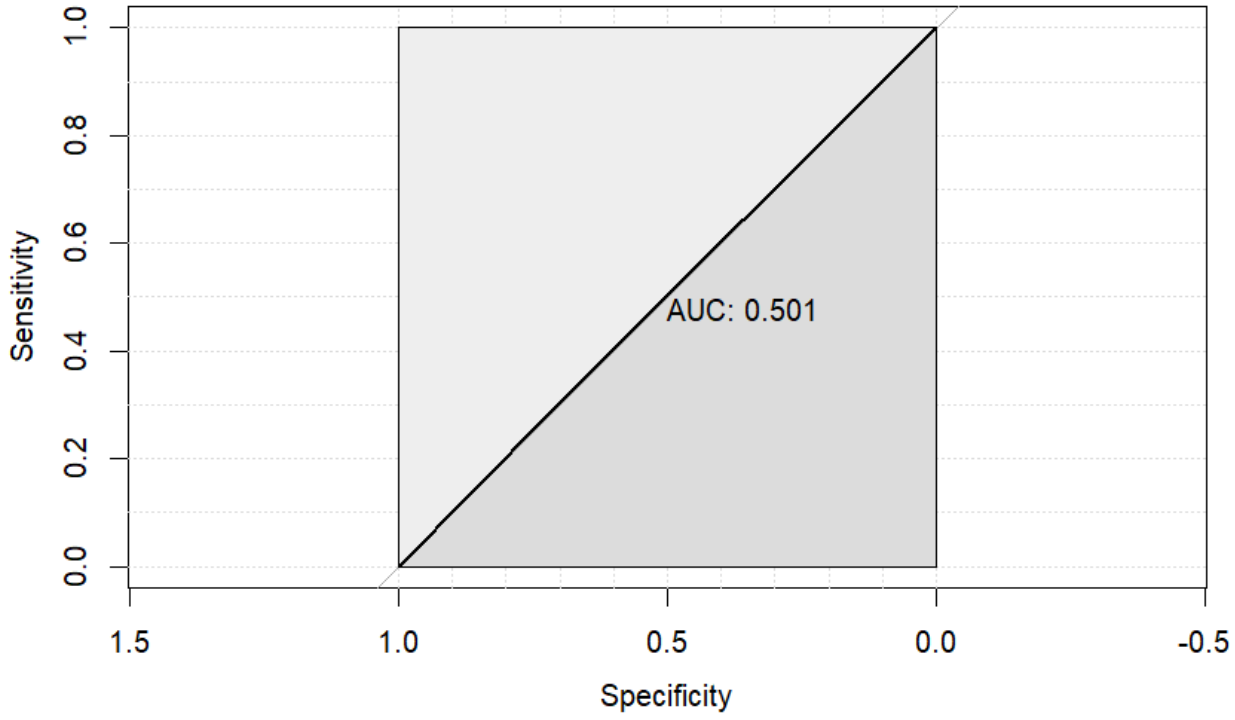


Figure 9 – Lasso Performance

```
                Accuracy : 0.6833
                  95% CI : (0.5504, 0.7974)
    No Information Rate : 0.7167
    P-Value [Acc > NIR] : 0.76610

                   Kappa : 0.0306

 Mcnemar's Test P-Value : 0.02178

            Sensitivity : 0.11765
            Specificity : 0.90698
         Pos Pred Value : 0.33333
         Neg Pred Value : 0.72222
             Prevalence : 0.28333
         Detection Rate : 0.03333
   Detection Prevalence : 0.10000
      Balanced Accuracy : 0.51231
```

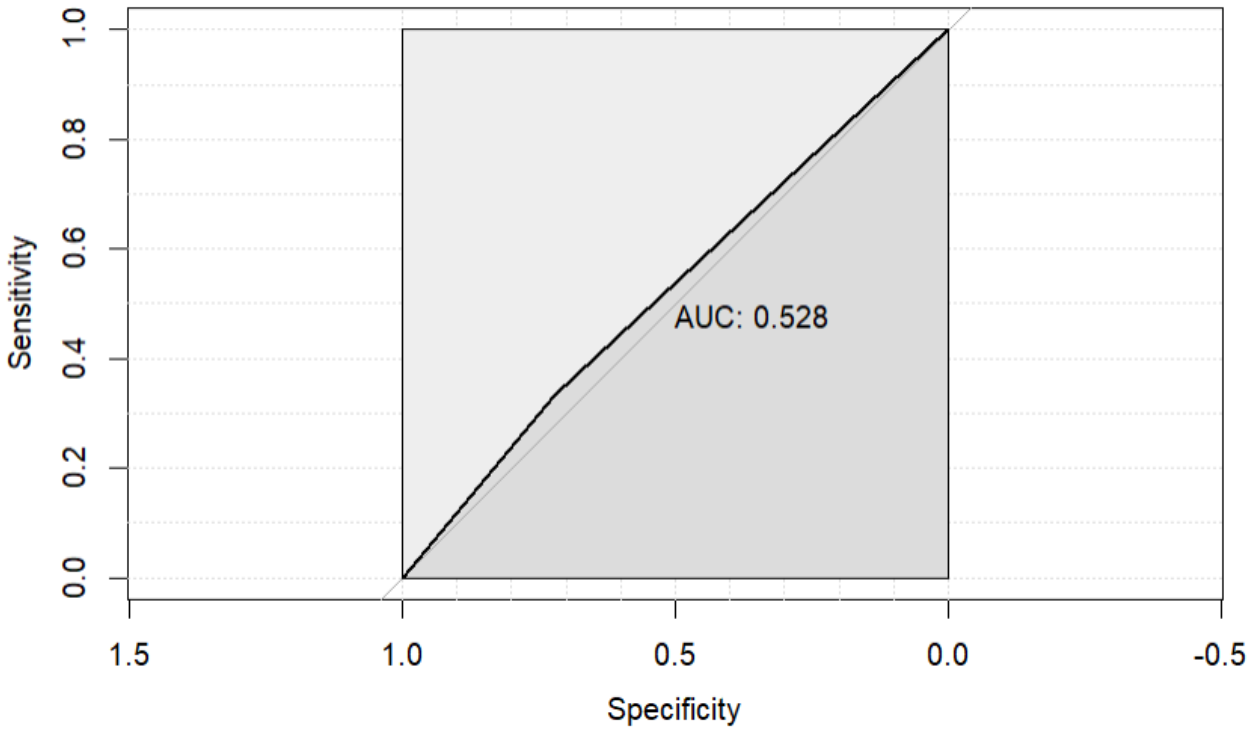


Figure 10 – Support Vector Machine Performance

```
                Accuracy : 0.75
                  95% CI : (0.6214, 0.8528)
     No Information Rate : 0.7167
     P-Value [Acc > NIR] : 0.340152

                   Kappa : 0.1993

  Mcnemar's Test P-Value : 0.001946

             Sensitivity : 0.17647
             Specificity : 0.97674
          Pos Pred Value : 0.75000
          Neg Pred Value : 0.75000
              Prevalence : 0.28333
          Detection Rate : 0.05000
    Detection Prevalence : 0.06667
       Balanced Accuracy : 0.57661
```

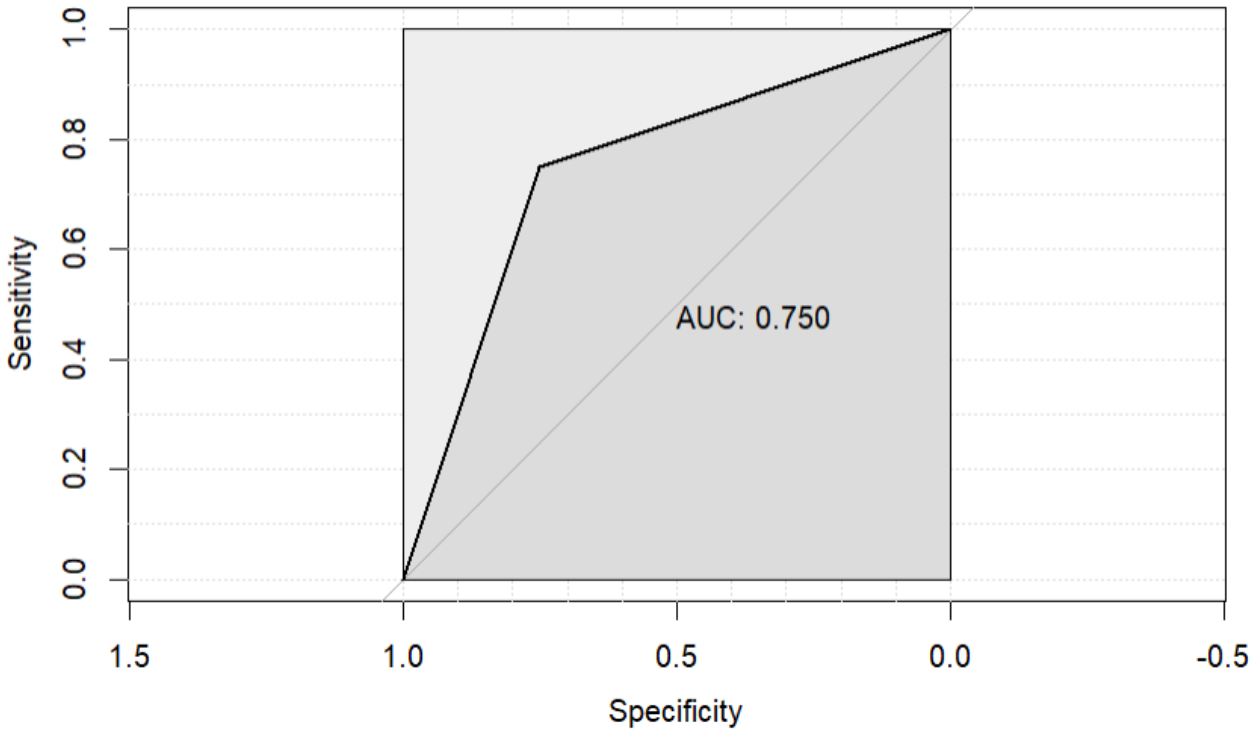


Figure 11 – Linear Discriminant Analysis Performance

```
                Accuracy : 0.7167
                  95% CI : (0.5856, 0.8255)
    No Information Rate : 0.7167
    P-Value [Acc > NIR] : 0.5649

                   Kappa : 0.2897

 Mcnemar's Test P-Value : 1.0000

            Sensitivity : 0.4706
            Specificity : 0.8140
         Pos Pred Value : 0.5000
         Neg Pred Value : 0.7955
             Prevalence : 0.2833
         Detection Rate : 0.1333
   Detection Prevalence : 0.2667
      Balanced Accuracy : 0.6423
```

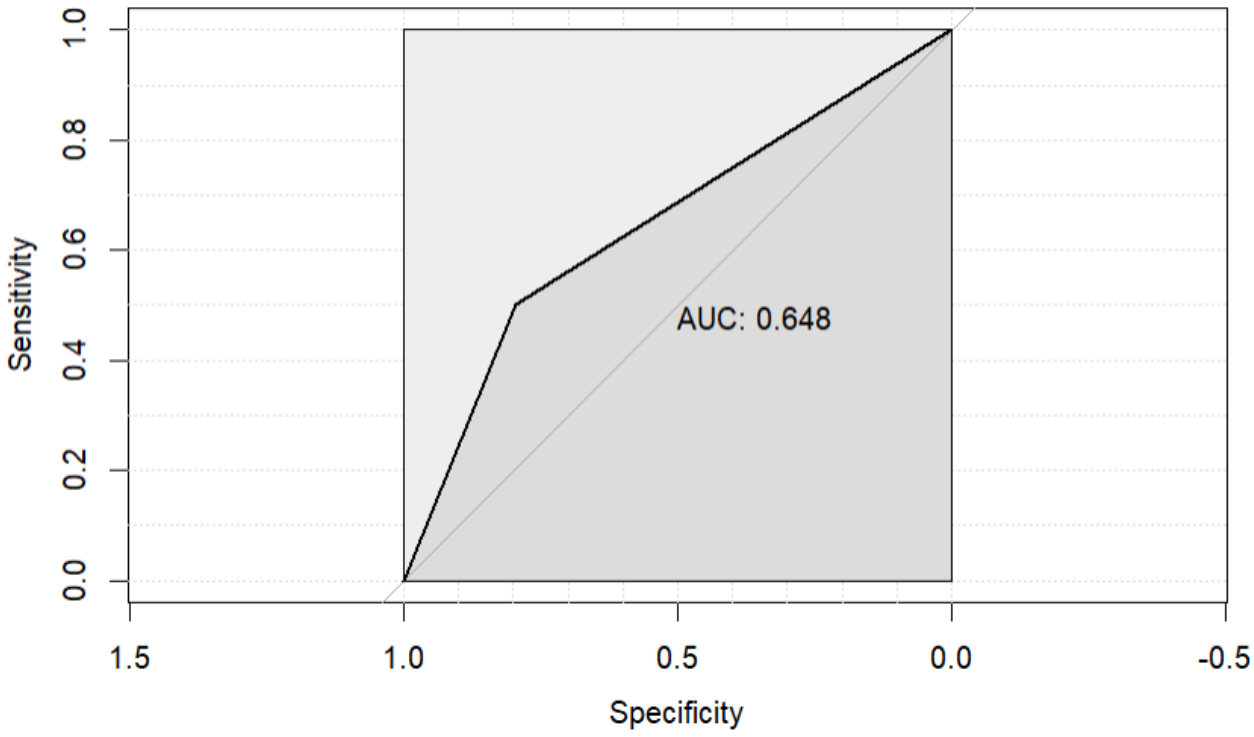


Figure 11.1 – Linear Discriminant Analysis Performance – Clinical Data

```
                Accuracy : 0.6667
                  95% CI : (0.5331, 0.7831)
     No Information Rate : 0.7167
     P-Value [Acc > NIR] : 0.8421

                   Kappa : 0.2074

  Mcnemar's Test P-Value : 0.8231

             Sensitivity : 0.4706
             Specificity : 0.7442
          Pos Pred Value : 0.4211
          Neg Pred Value : 0.7805
              Prevalence : 0.2833
          Detection Rate : 0.1333
    Detection Prevalence : 0.3167
       Balanced Accuracy : 0.6074
```

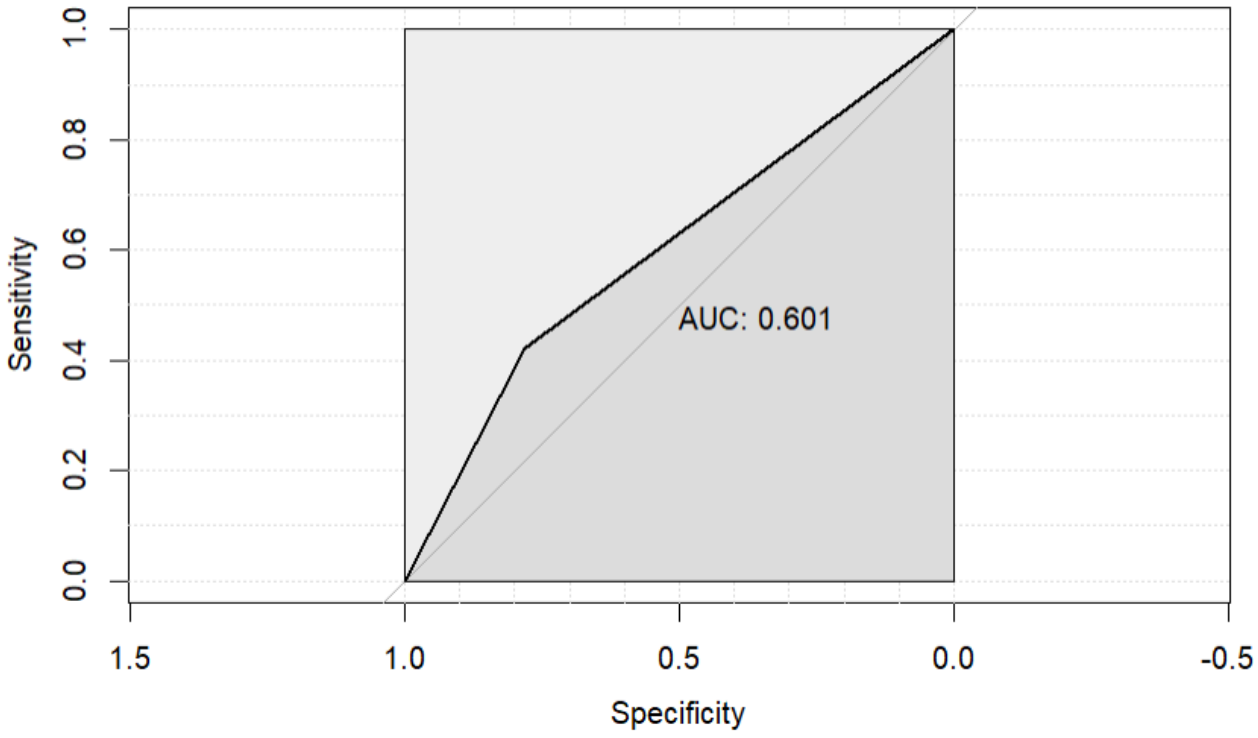


Figure 12 – Elastic Net Performance – Clinical Data

```
                Accuracy : 0.6667
                  95% CI : (0.5331, 0.7831)
    No Information Rate : 0.7167
    P-Value [Acc > NIR] : 0.8421

                   Kappa : 0.2074

 Mcnemar's Test P-Value : 0.8231

            Sensitivity : 0.4706
            Specificity : 0.7442
         Pos Pred Value : 0.4211
         Neg Pred Value : 0.7805
             Prevalence : 0.2833
         Detection Rate : 0.1333
   Detection Prevalence : 0.3167
      Balanced Accuracy : 0.6074
```

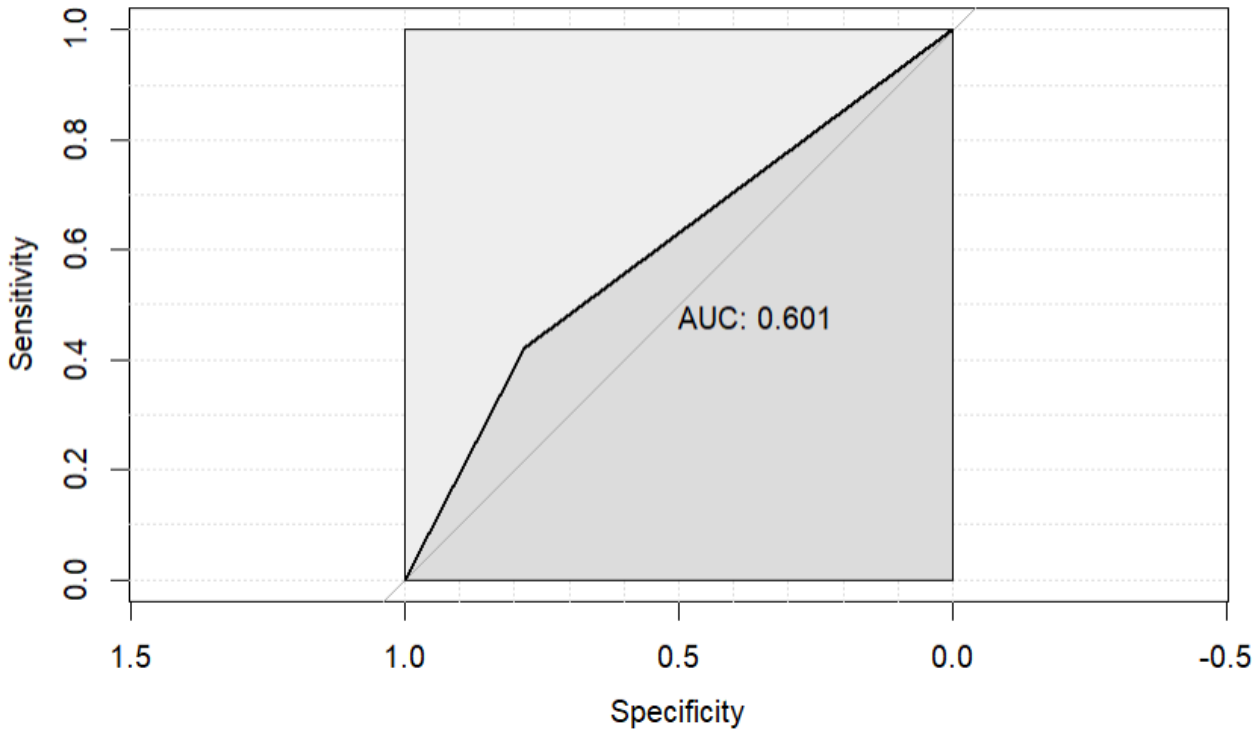


Figure 13 – Penalised Logistic Regression Performance – Clinical Data

```
                Accuracy : 0.7
                  95% CI : (0.5679, 0.8115)
    No Information Rate : 0.7167
    P-Value [Acc > NIR] : 0.6723

                   Kappa : 0.3103

 Mcnemar's Test P-Value : 0.4795

            Sensitivity : 0.5882
            Specificity : 0.7442
         Pos Pred Value : 0.4762
         Neg Pred Value : 0.8205
             Prevalence : 0.2833
         Detection Rate : 0.1667
   Detection Prevalence : 0.3500
      Balanced Accuracy : 0.6662
```

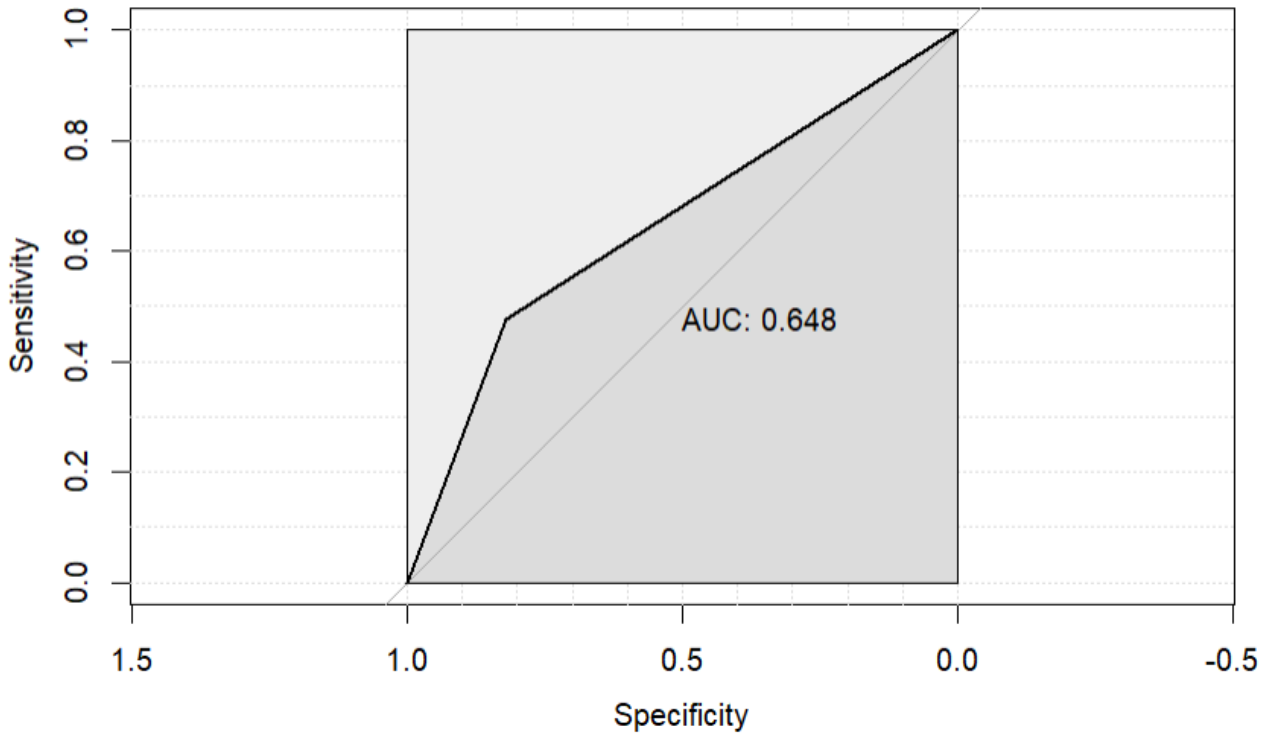


Figure 14 – Support Vector Machine Performance – Clinical Data

```
                Accuracy : 0.6667
                  95% CI : (0.5331, 0.7831)
    No Information Rate : 0.7167
    P-Value [Acc > NIR] : 0.8421

                   Kappa : 0.0812

 Mcnemar's Test P-Value : 0.2636

            Sensitivity : 0.23529
            Specificity : 0.83721
         Pos Pred Value : 0.36364
         Neg Pred Value : 0.73469
             Prevalence : 0.28333
         Detection Rate : 0.06667
   Detection Prevalence : 0.18333
      Balanced Accuracy : 0.53625
```

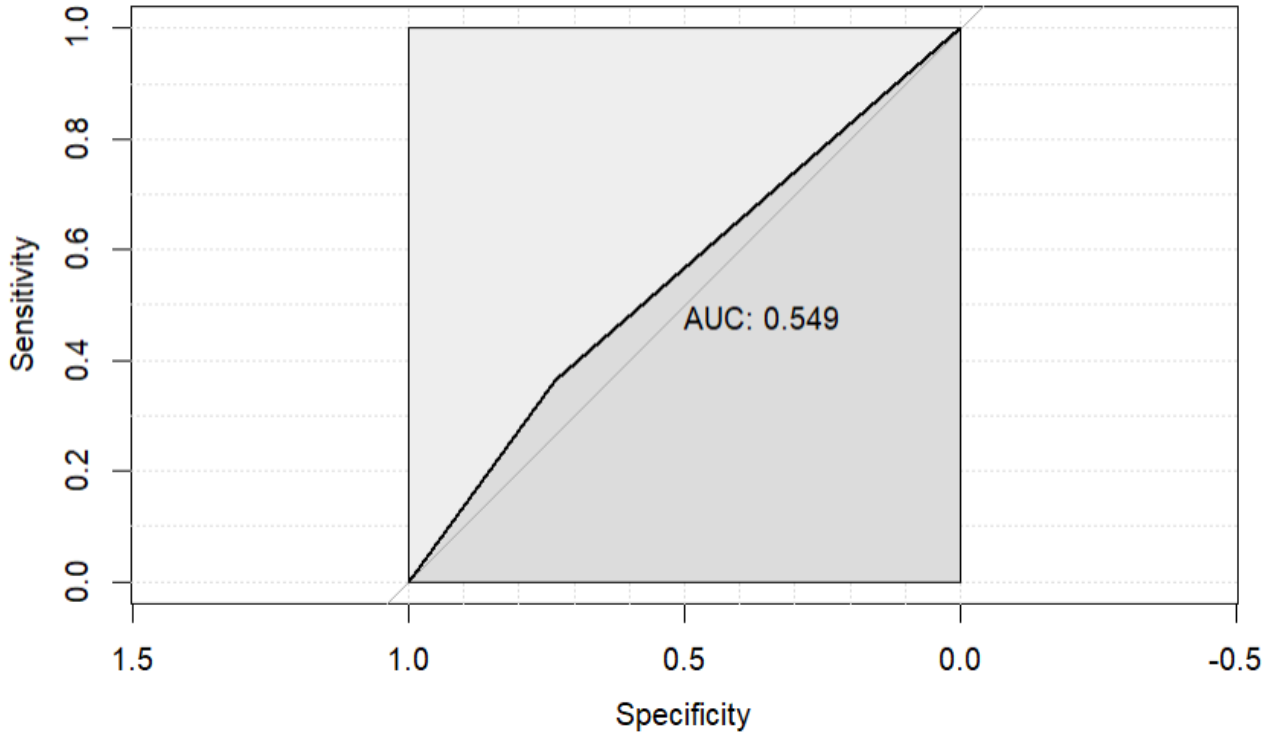


Figure 15 – Random Forest & Penalised Logistic Regression

```
                Accuracy : 0.75
                  95% CI : (0.6214, 0.8528)
    No Information Rate : 0.7167
    P-Value [Acc > NIR] : 0.340152

                   Kappa : 0.2347

 Mcnemar's Test P-Value : 0.009823

            Sensitivity : 0.23529
            Specificity : 0.95349
         Pos Pred Value : 0.66667
         Neg Pred Value : 0.75926
             Prevalence : 0.28333
         Detection Rate : 0.06667
   Detection Prevalence : 0.10000
      Balanced Accuracy : 0.59439
```

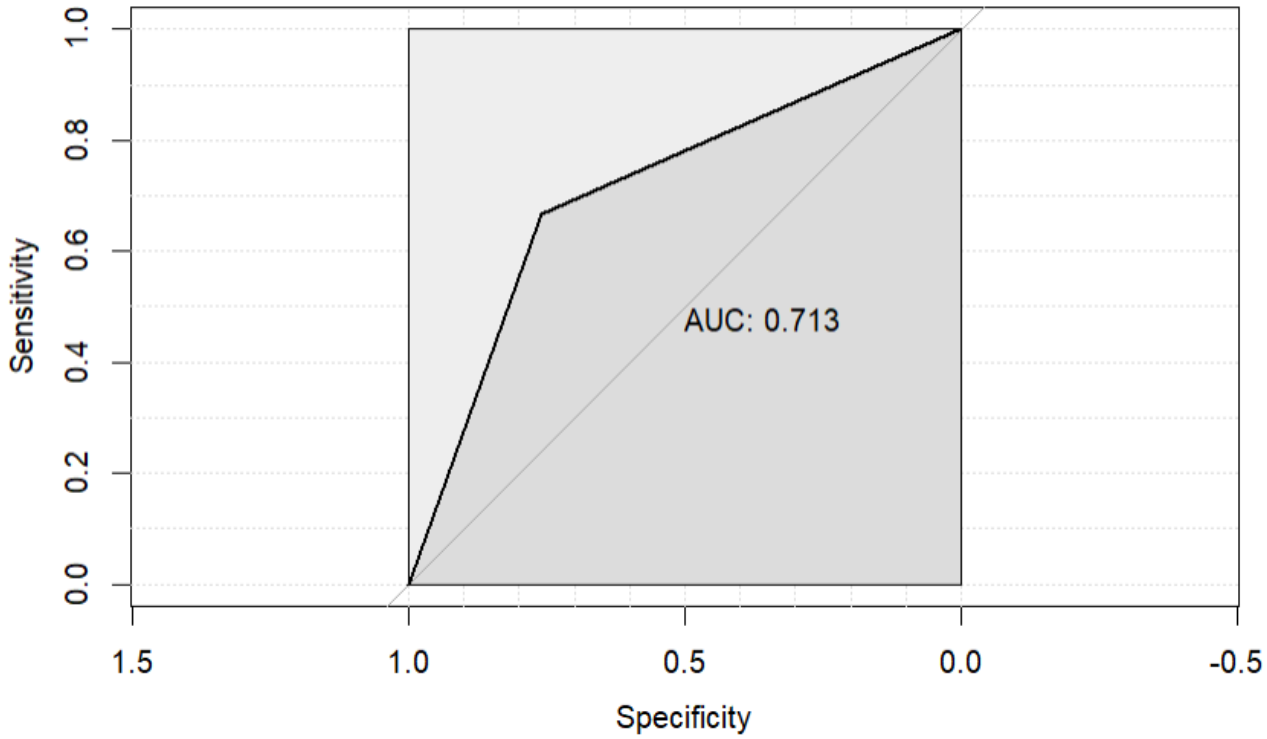


Figure 16 – Random Forest & Elastic Net

```
                Accuracy : 0.7167
                  95% CI : (0.5856, 0.8255)
    No Information Rate : 0.7167
    P-Value [Acc > NIR] : 0.564916

                   Kappa : 0.0925

 Mcnemar's Test P-Value : 0.003609

            Sensitivity : 0.11765
            Specificity : 0.95349
         Pos Pred Value : 0.50000
         Neg Pred Value : 0.73214
             Prevalence : 0.28333
         Detection Rate : 0.03333
   Detection Prevalence : 0.06667
      Balanced Accuracy : 0.53557
```

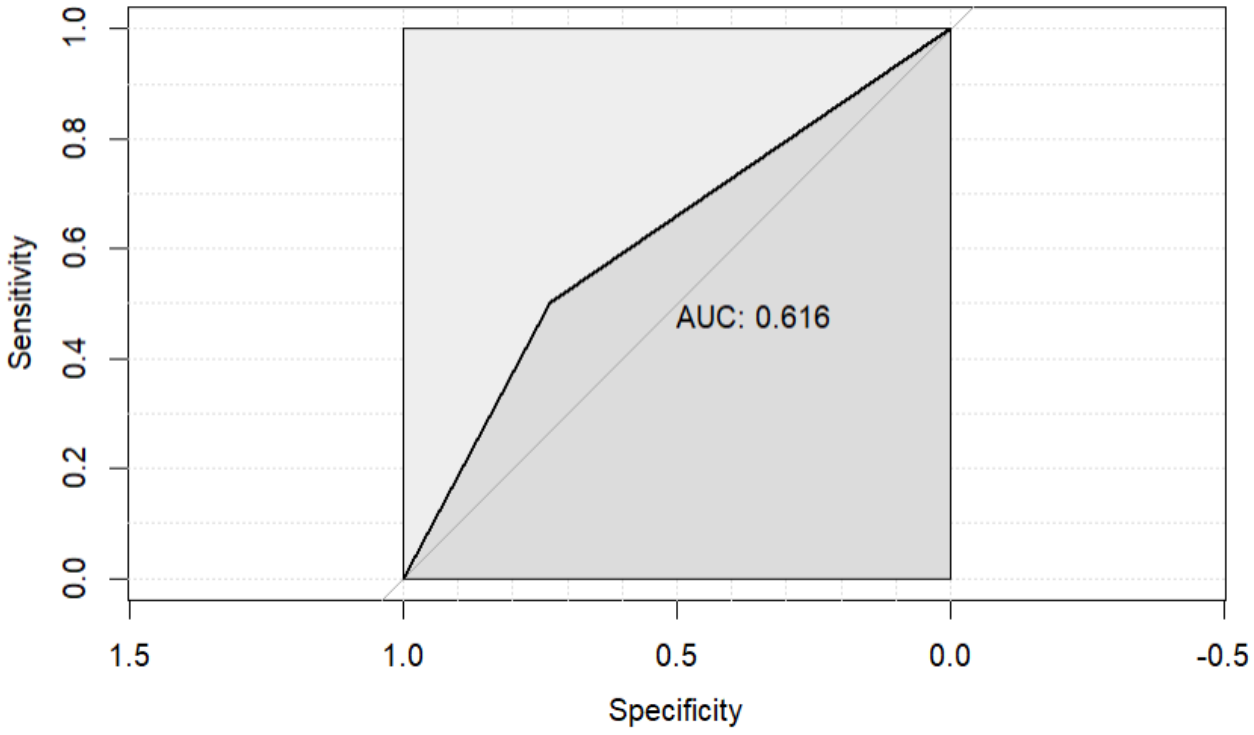


Figure 17 – Random Forest & Linear Discriminant Analysis

```
                Accuracy : 0.7167
                  95% CI : (0.5856, 0.8255)
    No Information Rate : 0.7167
    P-Value [Acc > NIR] : 0.564916

                   Kappa : 0.0925

 Mcnemar's Test P-Value : 0.003609

            Sensitivity : 0.11765
            Specificity : 0.95349
         Pos Pred Value : 0.50000
         Neg Pred Value : 0.73214
             Prevalence : 0.28333
         Detection Rate : 0.03333
   Detection Prevalence : 0.06667
      Balanced Accuracy : 0.53557
```

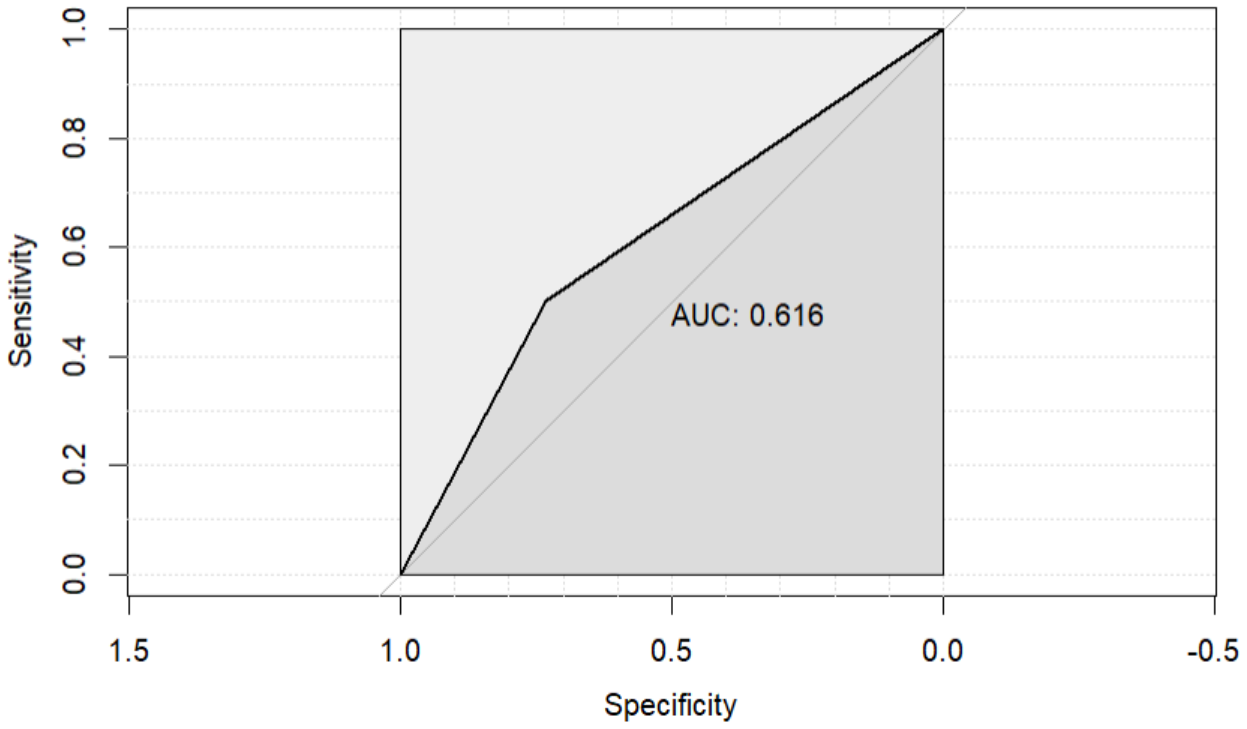


Figure 18 – Random Forest & Support Vector Machine

```
                Accuracy : 0.7167
                  95% CI : (0.5856, 0.8255)
     No Information Rate : 0.7167
     P-Value [Acc > NIR] : 0.564916

                   Kappa : 0.0485

Mcnemar's Test P-Value : 0.000685

            Sensitivity : 0.05882
            Specificity : 0.97674
         Pos Pred Value : 0.50000
         Neg Pred Value : 0.72414
             Prevalence : 0.28333
         Detection Rate : 0.01667
   Detection Prevalence : 0.03333
      Balanced Accuracy : 0.51778
```

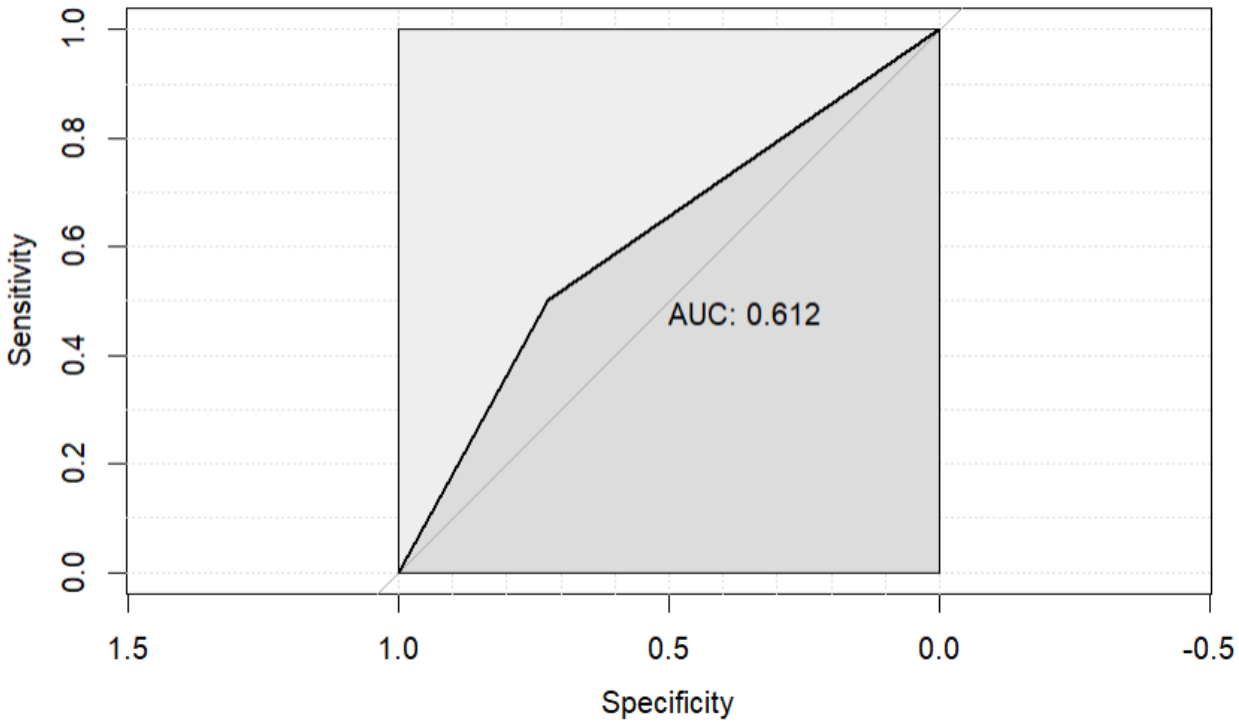


Figure 19 – Lasso & Penalised Logistic Regression

```
                Accuracy : 0.7
                  95% CI : (0.5679, 0.8115)
    No Information Rate : 0.7167
    P-Value [Acc > NIR] : 0.672274

                   Kappa : 0.0164

 Mcnemar's Test P-Value : 0.002183

            Sensitivity : 0.05882
            Specificity : 0.95349
         Pos Pred Value : 0.33333
         Neg Pred Value : 0.71930
             Prevalence : 0.28333
         Detection Rate : 0.01667
   Detection Prevalence : 0.05000
      Balanced Accuracy : 0.50616
```

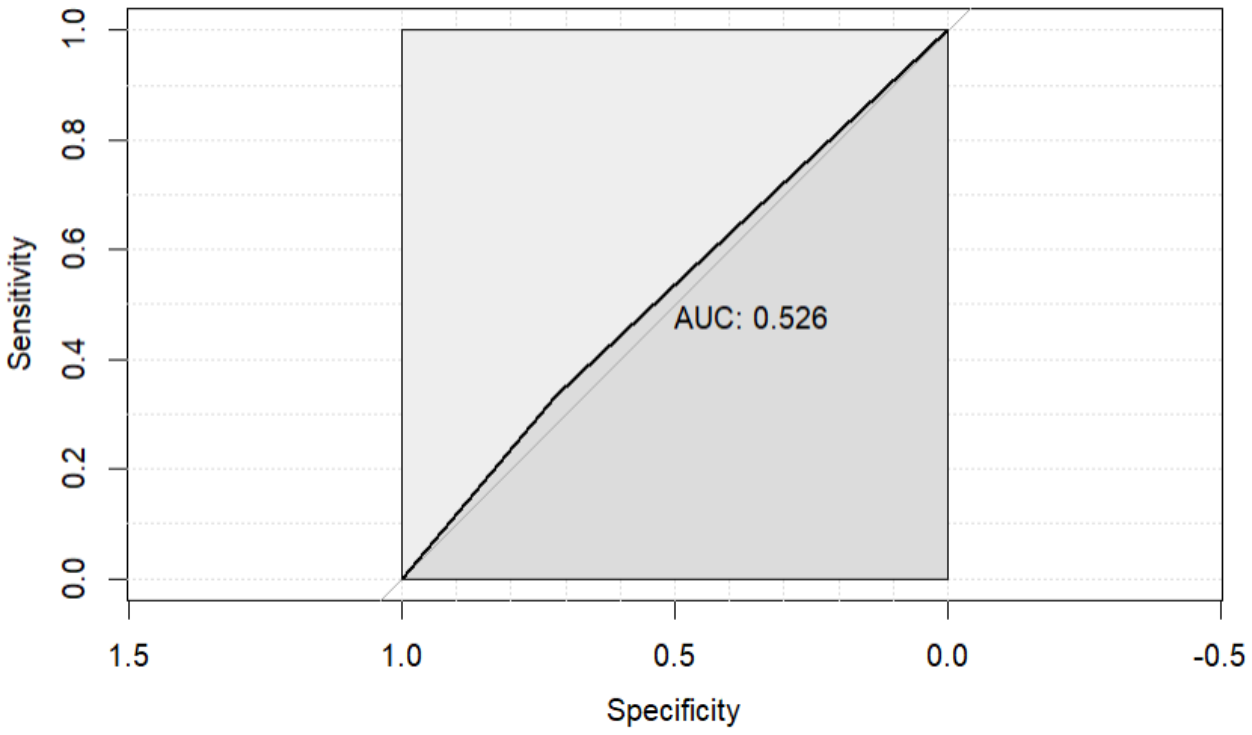


Figure 20 – Elastic Net & Penalised Logistic Regression

```
                Accuracy : 0.7
                  95% CI : (0.5679, 0.8115)
    No Information Rate : 0.7167
    P-Value [Acc > NIR] : 0.672274

                   Kappa : 0.0164

 Mcnemar's Test P-Value : 0.002183

            Sensitivity : 0.05882
            Specificity : 0.95349
         Pos Pred Value : 0.33333
         Neg Pred Value : 0.71930
             Prevalence : 0.28333
         Detection Rate : 0.01667
   Detection Prevalence : 0.05000
      Balanced Accuracy : 0.50616
```

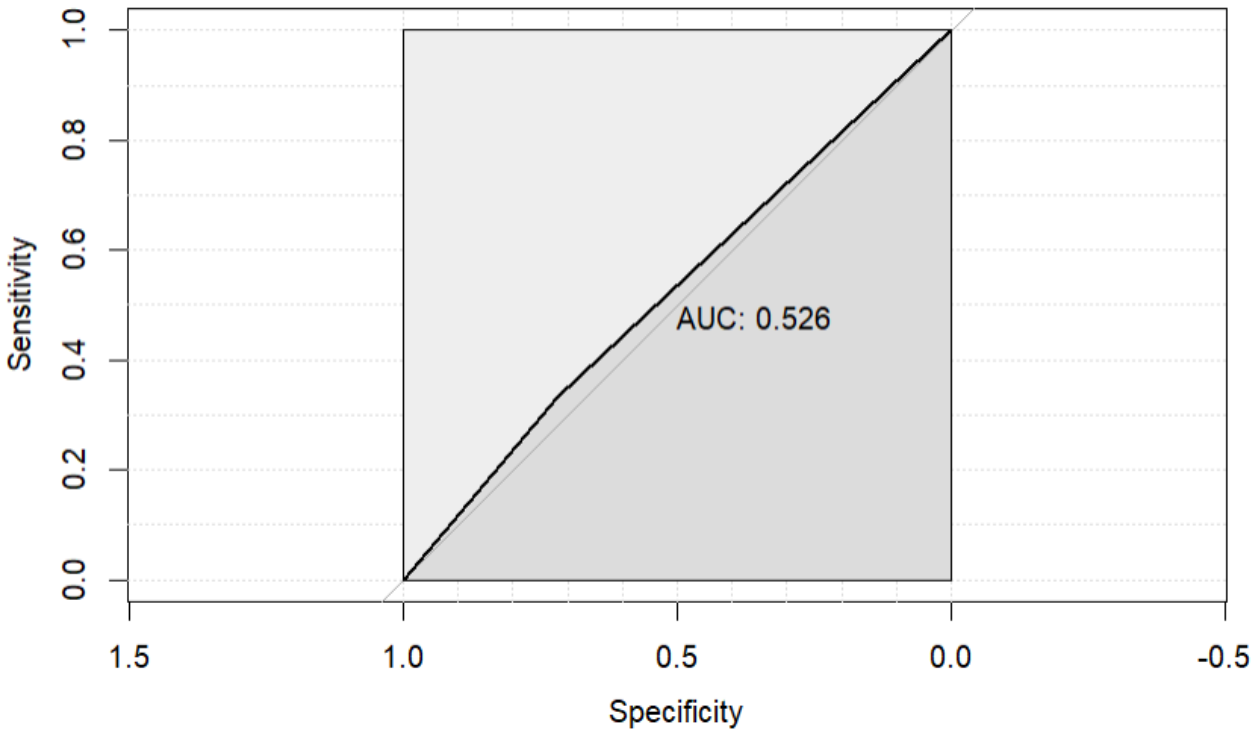


Figure 21 – Support Vector Machine & Penalised Logistic Regression

```
                Accuracy : 0.7333
                  95% CI : (0.6034, 0.8393)
    No Information Rate : 0.7167
    P-Value [Acc > NIR] : 0.451242

                   Kappa : 0.1257

 Mcnemar's Test P-Value : 0.001154

            Sensitivity : 0.11765
            Specificity : 0.97674
         Pos Pred Value : 0.66667
         Neg Pred Value : 0.73684
             Prevalence : 0.28333
         Detection Rate : 0.03333
   Detection Prevalence : 0.05000
      Balanced Accuracy : 0.54720
```

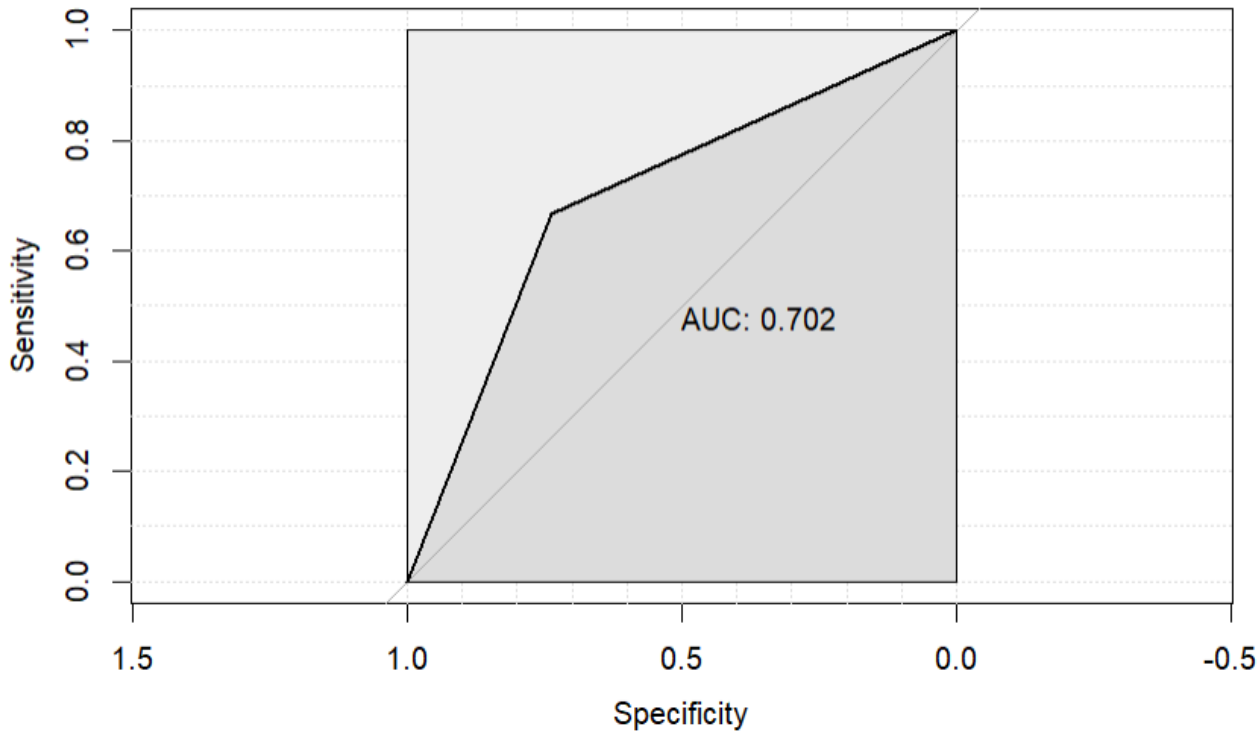


Figure 22 – Support Vector Machine & Elastic Net

```
                Accuracy : 0.7167
                  95% CI : (0.5856, 0.8255)
    No Information Rate : 0.7167
    P-Value [Acc > NIR] : 0.564916

                   Kappa : 0.0485

 Mcnemar's Test P-Value : 0.000685

            Sensitivity : 0.05882
            Specificity : 0.97674
         Pos Pred Value : 0.50000
         Neg Pred Value : 0.72414
             Prevalence : 0.28333
         Detection Rate : 0.01667
   Detection Prevalence : 0.03333
      Balanced Accuracy : 0.51778
```

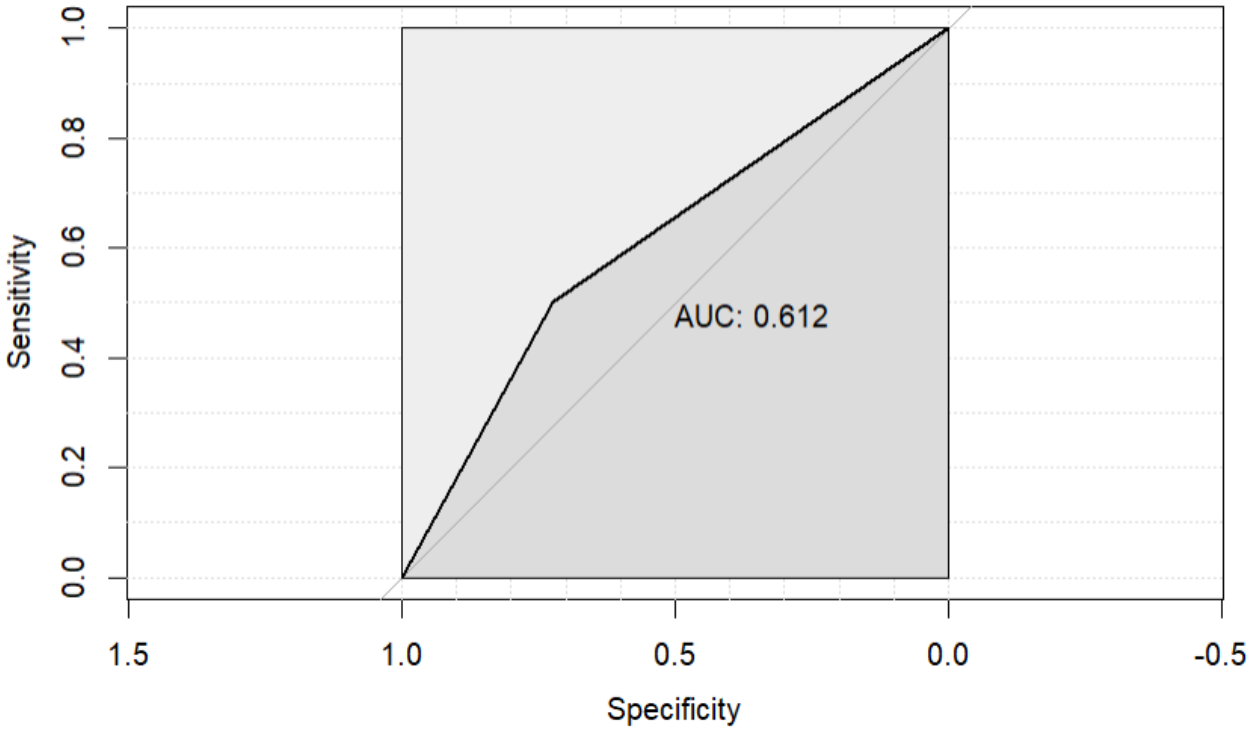


Figure 23 – Results Table

Row Labels	Average of Accuracy	Average of Balanced Accuracy	Average of Sensitivity	Average of Specificity	Min of McNemar's test p-value	Average of AUC
SVM on G	0.75	0.5777	0.1765	0.9767	0.0019	0.75
RF, PLR	0.75	0.5944	0.2343	0.9535	0.009823	0.713
SVM, PLR	0.7333	0.5472	0.1177	0.9767	0.001154	0.7018
Random Forest on G	0.7333	0.6	0.2941	0.907	0.08012	0.6601
PLR on C	0.7	0.6662	0.5882	0.7442	0.4795	0.6484
LDA on G	0.7167	0.6423	0.4706	0.814	1	0.6477
RF, LDA	0.7167	0.5356	0.1177	0.9535	0.003609	0.6161
RF, EN	0.7167	0.5944	0.1177	0.9535	0.003609	0.6161
SVM, EN	0.7167	0.5178	0.0588	0.9767	0.000685	0.6121
RF, SVM	0.7167	0.5178	0.0588	0.9767	0.000685	0.6121
SVM, LDA	0.7167	0.5178	0.0588	0.9767	0.000685	0.6121
LDA on C	0.6667	0.6074	0.4706	0.7442	0.8231	0.6008
SVM on C	0.6667	0.5363	0.2353	0.8372	0.2636	0.5492
Lasso on G	0.6833	0.5123	0.1177	0.907	0.02178	0.5278
LA, EN	0.7	0.5062	0.0588	0.9535	0.002183	0.5263
EN, PLR	0.7	0.5062	0.0588	0.9535	0.002183	0.5263
LA, LDA	0.7	0.5062	0.0588	0.9535	0.002183	0.5263
EN, EN	0.7	0.5062	0.0588	0.9535	0.002183	0.5263
EN, LDA	0.7	0.5062	0.0588	0.9535	0.002183	0.5263
LA, PLR	0.7	0.5062	0.0588	0.9535	0.002183	0.5263
Elastic Net on G	0.667	0.5007	0.1177	0.8837	0.04	0.5013
Elastic Net on C	0.6667	0.6074	0.4706	0.7442	0.8231	0.5013
Grand Total	**0.705327273**	**0.550568182**	**0.18445**	**0.911204545**	**0.000685**	**0.592168182**