Assessing Machine Learning Algorithms in Effective Temperature Prediction of Stellar Objects and Finding New Candidates for High and Low Mass and Cataclysmic Variable X-Ray Binary Systems

Joseph Patrick Allen

April 23, 2024

Contents

Abstract

This report assessed the performance of various linear, ensemble and neural networks algorithms, covering a broad range of complexity and interpret-ability in estimating the effective temperatures of known objects within GaiaDR3 from observational parameters, as well as to identify new candidates for high mass, low mass x-ray binaries and cataclysmic variables from a joint GaiaDR3 and eRASS1 catalogue. I ultimately concluded on the suitability of these algorithms for predicting effective temperatures in the GaiaDR3 dataset and presented a finalised list of candidates for high mass, low mass x-ray binaries and cataclysmic variables from joint GaiaDR3 and eRASS1 catalogues.

1 Introduction

The motivation for this project stemmed from the challenges posed by manual data analysis in the field of Astronomy, which has rapidly evolved into a 'Big Data' science [1]. Machine learning (ML) technologies, drawing inspiration from the human brain's pattern recognition capabilities, have become essential for managing the vast volumes of data generated by modern photometric, spectroscopic, and time-domain surveys. These datasets can range from tens of thousands to billions of measurements, making traditional interactive analysis with human oversight increasingly impractical [2]. ML algorithms are not only faster but also more efficient at handling large-scale data challenges, including noise and high dimensionality, thus enabling the rapid estimation of astrophysical parameters like mass, brightness, and temperature. Furthermore, they play a crucial role in identifying anomalies, paving the way for new discoveries in Astronomy. As computational power continues to expand, the application of ML and data science is expected to drive further innovations and breakthroughs in the field [3].

1.1 What is Machine Learning?

ML is a branch of artificial intelligence that employs algorithms and statistical models to replicate the way humans learn, involving tasks such as decision-making and prediction based on data [4]. ML is primarily divided into three types: supervised, unsupervised, and reinforcement learning. In supervised learning, models are trained using labelled datasets to learn and then predict outcomes on new, unseen data. Unsupervised learning involves training models on unlabelled data to discover hidden patterns or structures. Reinforcement learning focuses on making decisions through trial and error in an environment to achieve a goal, learning from the outcomes of its actions. This report focuses only on supervised and unsupervised learning. The process of learning in ML includes three main stages: the decision process (making predictions or classifications), the error function (evaluating predictions), and model optimization (adjusting parameters to minimize errors and improve accuracy), repeating until the model reaches a satisfactory level of accuracy.

1.2 Regression Algorithms

The following algorithms are all sourced from https://scikit-learn.org/stable/ A regression problem in ML is a type of supervised learning problem where the goal is to predict a continuous output variable based on the mathematical relationship between one or more input variables [5]. The output variable can take any real number value, rather than being limited to a set of categories or classes, which I discuss in classification. The target variable can take any value within a range, such as an effective temperature, radial velocity, metallicity etc. The regression model learns a function that maps input features to the predicted output value. The performance of such is typically evaluated using a loss function that measures the difference between the actual values and the predictions. These include but not limited to, Mean Squared Error (MSE) and Mean Absolute Error (MAE) [6].

1.2.1 Linear Regression

Linear regression is one of the most fundamental statistical methods used in machine learning and data analysis for predicting a continuous outcome variable based on one or more input variables. The goal of linear regression is to find the best-fit line that represents the relationship between the input features and the output [7]. This relationship is modelled as a straight line, which is represented by the equation: $y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + ... + \beta_n x_n + \epsilon$

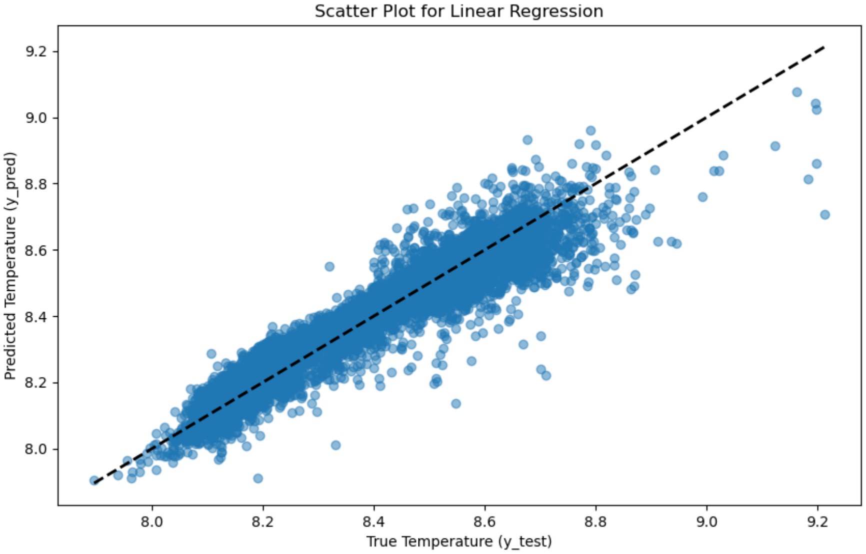


Figure 1: The error between predicted and actual temperatures, demonstrated by an overlaying line of gradient unity.

1.2.2 Random Forest Regressor

The idea behind Random Forest Regressor (RFR) is to create a set of decision trees from a randomly selected subset of the data and then aggregate their predictions to make a final prediction. This process is known as bootstrapping and aggregation [8]. For each decision tree in the forest, each sample is drawn with replacement, and the size of the sample is the same as the size of the original dataset. This process introduces 'randomness' and helps in reducing overfitting by providing different views of the data to each decision tree [9]. At each node of the decision tree, a subset of features is randomly selected, further introducing randomness and helps in preventing any single feature from dominating the model [10]. The tree is then grown to the maximum depth or until a stopping criterion is met [11]. The final prediction of RFR is obtained by averaging the predictions of all the decision trees in the forest. This step combines the predictions of multiple decision trees to produce a more accurate and stable prediction.

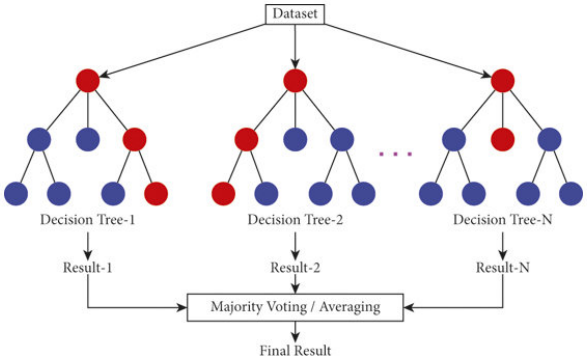


Figure 2: The aggregation of the predictions of multiple decision trees. Note that in this case, the majority voting refers to Random Forest Classifier which is discussed later [12].

The number of decision trees can be specified, a larger number of trees could lead to better performance but also substantially increases the running time and the risk of overfitting. Criterion is the function that measures the quality of a split in the decision tree. Depth of trees can be altered to capture more complex patterns in the data but also increases the risk of overfitting. The minimum number of samples required for a node to be split can be given to lead to a more robust model but also at the risk of underfitting, while the minimum number of samples required to be at a node can be stated alongside number of samples to suit the dataset. Often, a grid search is implemented to find the optimal combination and value of the hyperparameters [13].

1.2.3 MLPRegressor

MLPRegressor is a type of feedforward artificial neural network [14] that includes at least three layers of neurons: an input, hidden layer, and an output layer. Each neuron in one layer connects with a certain weight to every neuron in the following layer. Each neuron in the hidden layer applies a non-linear activation function (typically a sigmoid function, a hyperbolic tangent (tanh), or a ReLU (Rectified Linear Unit)) to the weighted sum of its inputs [15]. The output layer has a single neuron that outputs the predicted continuous value which is then passed through a linear function to produce the final output.

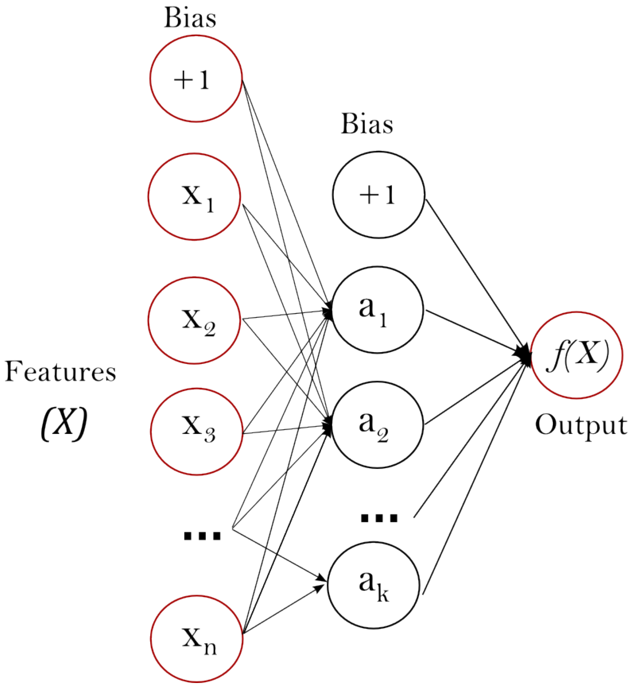


Figure 3: On the left is the input layer, containing a set of neurons representing the input features. Each neuron in the hidden layer transforms the values from the previous layer with a weighted linear summation, followed by a non-linear activation function. The output layer receives the values from the last hidden layer and transforms them into a continuous variable [16]. The goal of training the MLP is to find the optimal weights, biases, and activation function parameters that minimise the difference between the predicted and the actual output.

The most important hyperparameter to consider is the size of the hidden layers. Trialling of the number of neurons per layer and number of layers is paramount to providing the most flexibility of the model to the data whilst not overfitting the training data, such the model won't generalise to unseen data well. Alpha is a regularisation parameter that controls the amount of L2 regularization applied to the weights of the network. The learning rate controls the step size at each iteration of the optimisation algorithm. It determines how much the weights of the network are updated in response to the calculated error at each step of the training process. The latter is critical as if the learning rate is too high, the model might overshoot and fail to converge. If it's too low, the model might take too long to converge or get stuck in a suboptimal solution (local minima) [17].

1.3 Classification Algorithms

A classification problem involves predicting a discrete output such as an integer, boolean, or string [18] from several input features, with performance evaluated using a classification report. This gives insight into the precision (the accuracy of positive predictions), recall (the ability to capture all positive instances), f1-score (a harmonic mean of precision and recall), and support (the occurrence of each class).

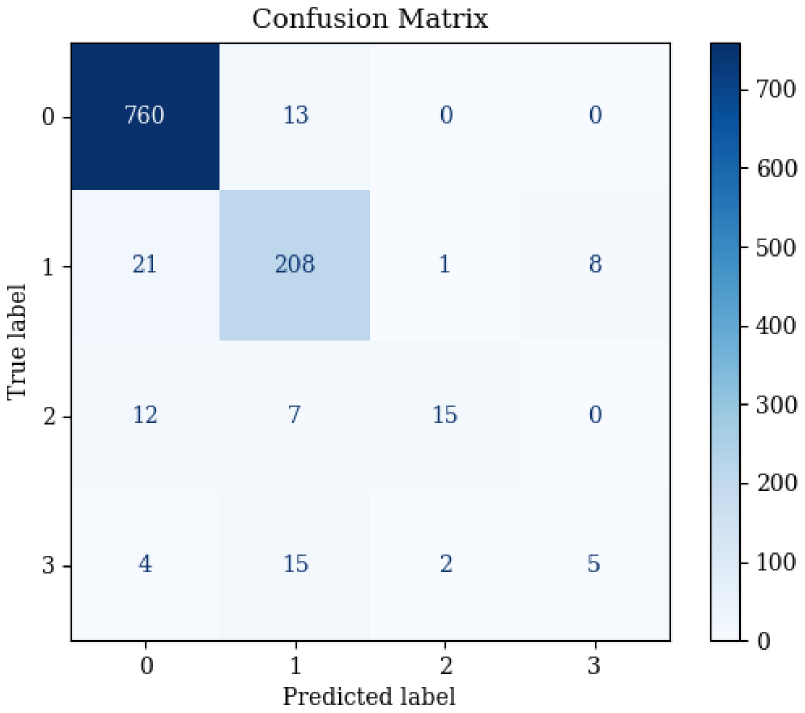


Figure 4

A Multiclass Confusion Matrix is an extension of the Binary Confusion Matrix in which illustrates the results of the logistic regression by plotting the predictions of each class against the actual numbers of each class in a

4x4 grid. This way we can identify true positives in the leading diagonal and instances of misdiagnosis (false negatives and false positives) in the off diagonals [19].

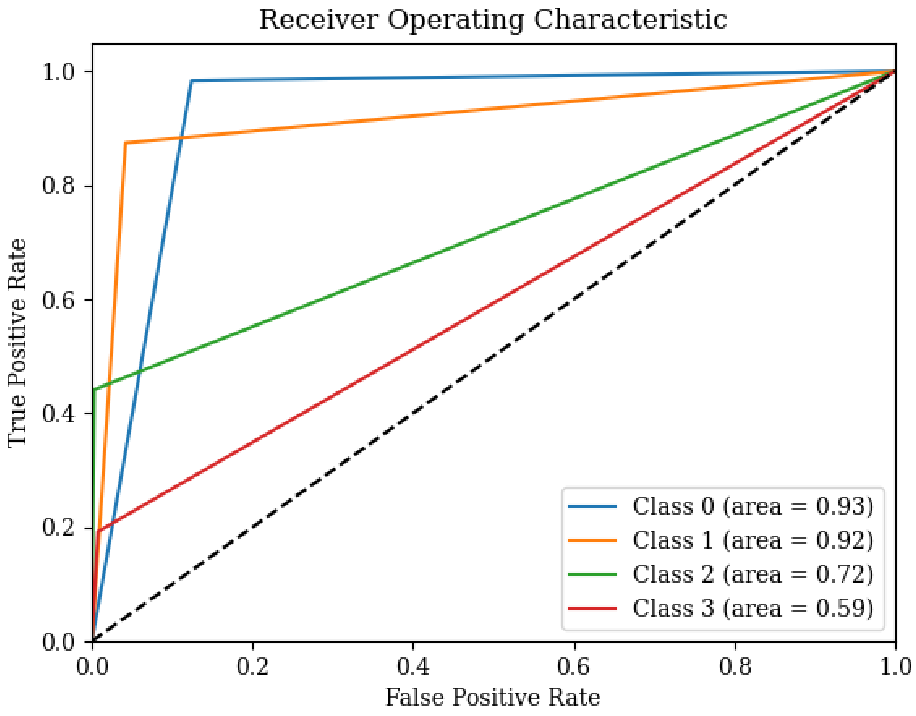


Figure 5

A Multi-Class Receiver Operating Characteristic (ROC) curve plots the true positive rate (TPR) against the false positive rate (FPR) for various threshold settings, which determine how a model classifies instances as positive or negative [20]. The TPR is the proportion of actual positives correctly identified, while the FPR is the proportion of negatives incorrectly labelled as positives. The area under the curve (AUC) evaluates the model's ability to distinguish between classes, with 1 indicating perfect performance, 0.5 implying no better than random guessing, and values between 0.5 and 1 suggesting improvement over random. A model with lines above the diagonal on this curve indicates better performance, showing a stronger true positive rate compared to the false positive rate, thus highlighting its effectiveness in differentiating between classes. This analysis shows that the logistic regression model particularly excels in classifying class 0 and 1.

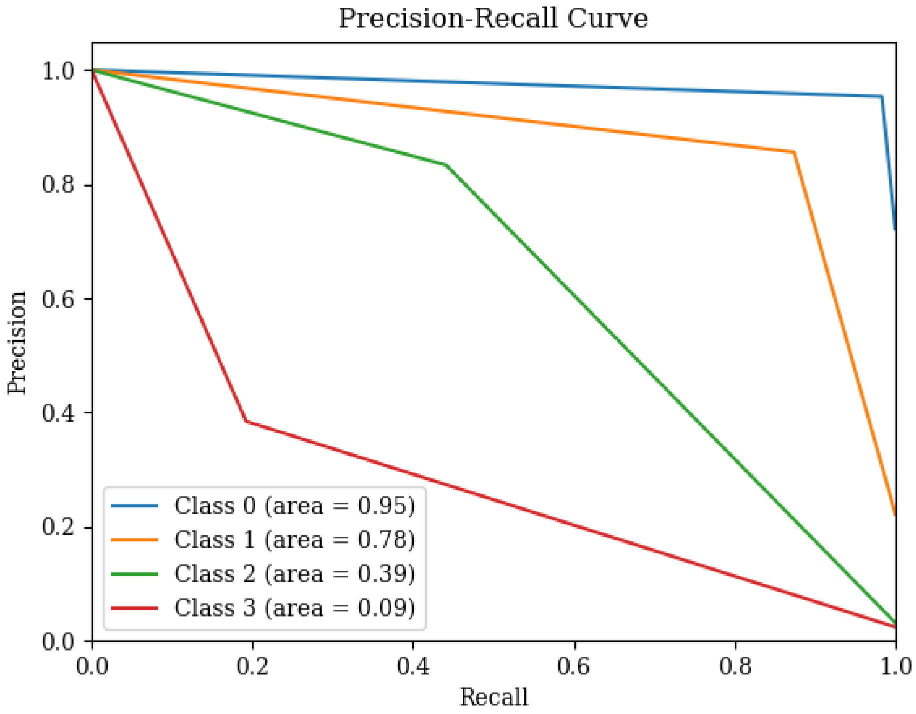


Figure 6

The Multi Class Precision-Recall curve pictured above shows the trade-off between precision and recall for different thresholds. A high area under the curve represents both high recall and high precision, where high precision relates to a low false-positive rate, and high recall relates to a low false-negative rate. A model with perfect precision and recall would have a curve that goes straight up the y-axis (recall) and then along the x-axis (precision). The closer the curve follows this path, the better the classifier's performance [21]. However, in practice, there is often a trade-off between precision and recall. A model that is trying very hard to avoid false negatives will likely classify many negative instances as positive, decreasing its precision. Conversely, a model that aims to maintain high precision will likely miss some positive instances, reducing its recall. Ideally the researcher should plot this only if the picture painted aids in understanding why the classifier has performed in such a way.

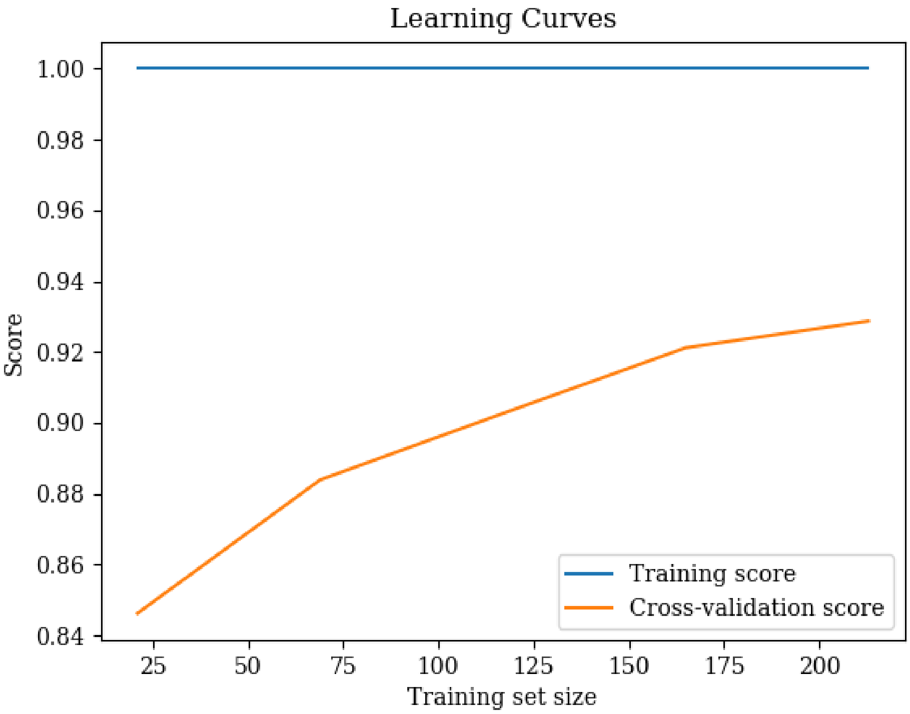


Figure 7

A learning curve plots the size of the training set against a model's performance on both training and validation sets [22], serving as an essential tool for evaluating how performance changes with additional data. This curve helps identify the optimal training set size by showing where training and cross-validation scores converge, indicating a balance between fitting the data and generalizing to new data. High training scores with low validation scores suggest overfitting, where the model learns noise rather than the data's actual pattern, while uniformly low scores indicate underfitting, where the model is too simplistic. Increasing the training set size can mitigate overfitting by providing a broader data spectrum, whereas underfitting may require a more complex model. The curve also indicates when more data no longer improves the model, suggesting it has reached its learning limit.

1.3.1 Logistic Regression

Logistic regression is a generalised linear model that handles classification problems with more than two classes. It uses a softmax function to convert a vector of real numbers representing the log-odds of each class into a probability distribution across the classes [23]. Each class's probability is modelled as the exponential of its log-odds normalized by the sum of exponentials of all class log-odds. The model's parameters, including an intercept and coefficients for each predictor variable, are determined through Maximum Likelihood Estimation (MLE), which seeks to maximize the likelihood of the observed class distribution. This approach not only models the probability of each class effectively but also allows for straightforward interpretation of how predictor variables influence class probabilities.

1.3.2 Random Forest Classifier

Random Forest Classifier (RFC) is the mirror classifier of RFR, and as such performs identically with the only differences being the performance metrics and discrete output. Note, RFC also concerns the same group of hyperparameters as RFR [24].

1.3.3 MLPClassifier

MLPClassifier is the mirror classifier to MLPRegressor, and as such, is another type of feedforward artificial neural network that implements a Multi-Layer Perceptron (MLP) for classification tasks. Again, this mirror algorithm concerns the same group of hyperparameters as MLPRegressor [25].

1.4 Feature Selection

Feature selection is a critical step in modelling both regression and classification problems, aiming to remove non-informative or redundant predictors to improve model efficiency and performance. This process is especially vital when dealing with many inputs, as excessive variables can degrade performance by including irrelevant features [26]. Feature selection techniques can be categorised into supervised and unsupervised methods. Supervised methods utilise the target variable to guide the selection process, identifying and excluding irrelevant variables. In contrast, unsupervised methods do not use the target variable. Some machine learning models integrate feature selection intrinsically within the learning process, known as intrinsic methods. While feature selection explicitly selects or excludes certain features, dimensionality reduction involves creating a new projection of the data, resulting in entirely new input features.

1.5 Principal Component Analysis

Principal Component Analysis (PCA) is a statistical technique that transforms possibly correlated variables into a set of linearly uncorrelated variables known as principal components, by using an orthogonal transformation. This method is often employed in classification and regression for dimensionality reduction, which may enhance model performance by lowering computational complexity and improving generalisation to unseen data. The process begins with standardizing the data to have zero mean and unit variance, ensuring each variable contributes equally to the analysis. A covariance matrix is then calculated to measure the variance and covariance between variables. This matrix is decomposed into eigenvectors (indicating the directions of the principal components) and eigenvalues (representing the variance explained by each component). The eigenvectors are ranked by their eigenvalues in descending order, and the top k eigenvectors, chosen by the researcher, define the new feature space. The original data is projected onto these principal components to obtain transformed data [27].

1.6 Best Practices

In ML applications for astronomy, it's crucial to ensure analyses on physical data are both valuable and avoid common pitfalls. ML implementations should meet three key criteria: accuracy, reproducibility, and usefulness, which ensure reliable results, consistent performance, and practical benefits over traditional methods [28]. These standards can be adjusted when justified by researchers. Using ML in astronomy aims to enhance computational efficiency, improve precision, enable collaboration, and address unique challenges. Evaluating ML against traditional methods is essential for establishing credibility within the astronomical community. Simplicity in models is preferred for easier interpretability and troubleshooting. Transparency is critical to prevent the pitfalls of complex 'black box' models, which can lead to issues like overfitting and poor generalization. Proper dataset preparation, choosing the right evaluation metrics, and detailed reporting are best practices in ML research for astronomy, with deviations requiring strong justification. Initially defining the target variable and data scope is essential to ensure fairness and clarity in model predictions.

1.7 Data Collection

This report uses the following publicly available datasets:

Gaia Data Release 3 (GaiaDR3) - Released by the ESA on June 13, 2022, GaiaDR3 offers an extensive catalogue of over a billion astronomical objects, detailing position, parallax, and proper motion. This release expands on previous data with new products for extended and non-single stars, including detailed stellar classifications and parameters (e.g., effective temperature, metallicity, G-band extinction) for 470 million objects [29].

eROSITA Data Release 1 (eRASS1) - Showcasing the first six months of the SRG/eROSITA all-sky survey, this release was made public on January 31, 2024, covering the Western Galactic hemisphere. It includes a comprehensive catalog of high-energy cosmic sources, significantly surpassing detections by long-standing observatories like XMM-Newton and Chandra [30].

Multiwavelength Catalogue of 10,000 4XMM-DR13 Sources with Known Classifications - This catalogue from the 4th XMM-Newton Serendipitous Source Catalogue includes 10,000 classified X-ray sources with multi-wavelength counterparts. It is aimed at facilitating the training of machine learning algorithms for classifying X-ray sources [31].

Catalogues of High-Mass and Low-Mass X-ray Binaries in the Galaxy - These compilations detail 152 high-mass and 339 low-mass X-ray binaries, providing positional data, spectral types, and radial velocities. Many of these sources are detected in recent GaiaDR3 updates. [32][33]

The Ritter-Kolb Catalogue - Contains detailed entries for 1429 cataclysmic binaries, including coordinates, magnitudes, orbital parameters, and stellar parameters. It also features a cross-reference list to help identify and study these objects [34]. Structure of Report

1.8 Structure of Report

In this report I assessed the performance various ML models, making sure to cover a broad range of complexity and interpretability to estimate the effective temperatures of known objects within GaiaDR3 from observational parameters, as well as to identify new candidates for high mass, low mass x-ray binaries and cataclysmic variables from a joint GaiaDR3 and eRASS1 catalogue. Each task commenced by referring to the above datasets and any justifications in preprocessing required to prime these datasets for ML, afterwards presenting the results of the regression/classification alongside key figures and metrics illustrating their performance. Once all methods had been considered, I ultimately concluded on the suitability of these algorithms for predicting effective temperatures in the GaiaDR3 dataset and presented a finalised list of candidates for high mass, low mass x-ray binaries and cataclysmic variables from joint GaiaDR3 and eRASS1 catalogues.

2 Effective Temperature Regression

2.1 Context

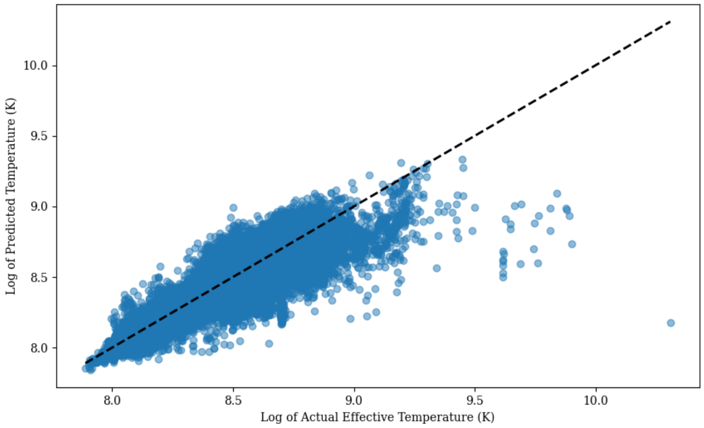
The effective temperature of a stellar object is calculated in GaiaDR3 using data from the Blue (BP) and Red (RP) Photometric bands, G-band magnitude, and parallax measurements [35]. This calculation involves the GSP-Phot module, which employs forward modelling and Markov Chain Monte Carlo (MCMC) techniques to estimate effective temperature, surface gravity, metallicity, and radius. Utilising ML algorithms to learn from known effective temperatures allows for increasingly accurate predictions as more data becomes available. This advancement enhances the precision of astronomical research by automating temperature estimations, thereby freeing astronomers to focus on more exploratory aspects of astrophysics [36]. Such optimisation of human and computational resources leads to more efficient project management. Good solutions should closely match the values from GaiaDR3 and include an error metric. Algorithms should employ a train-test split to prevent data leakage and use cross-validation to ensure model integrity on different subsets of the data [37]. Despite the challenges of explainable AI in complex models, solutions should strive for a degree of interpretability to enhance trust in the results. These algorithms should handle outliers and noise effectively, requiring suitable scaling to maintain feature balance. Additionally, effective solutions should explore feature selection and implement a grid search for optimal hyperparameters to maximise model performance [38].

2.2 Preprocessing

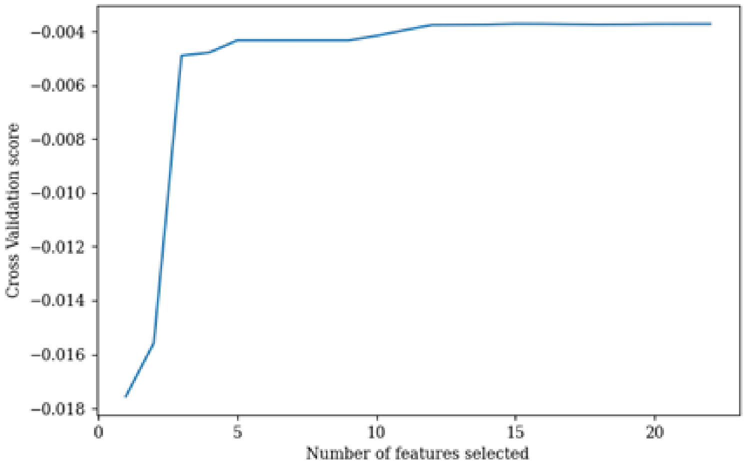
To prime GaiaDR3 for ML application, a one million subset of the original GaiaDR3 dataset was taken using TOPCAT [39]. Sheet rows containing empty entries were removed, additionally checks were implemented to ensure each parameter contained values of the same data type, such the algorithms could run without interruption. By consulting the GaiaDR3 documentation, I was able to choose purely observational parameters in right ascension, declination, parallax, proper motion, G-band flux, BP-band flux, RP-band flux. Magnitudes for the respective flux bands were used as well as BP-RP, G-RP, BP-G ratios. In the figure below is a corner plot of all mentioned inputs to demonstrate their dependencies on each other as well as the effective temperature. Note, that the strongest linear relationships exist for effective temperature and colour ratios, while for parallax and G-band magnitude the lack of a linear correlation inferred a more intricate dependence. Note that parallax, proper motion, G-band flux, BP-band flux, RP-band flux, and Teff are log scaled such that the wide range of datapoints would create unnecessary outliers for ML model training.

2.3 Results

Cross Validation Recursive Feature Elimination (RFECV) was used in a linear regression pipeline to recursively remove attributes and train linear regression across 5-fold subsets of the training data to identify the 5 most important observational parameters for the 320k testing set. It uses the negative mean squared error to identify which attributes degrade model performance most when left out. Linear regression is by far the most interpretable algorithm in its mathematical formulation, yet its inflexibility in a lack of hyperparameters and penalisation for choice of weights, made the algorithm too coarse to capture non-linear relationships. The 5 parameters highlighted were G-band flux, G-band magnitude, BP-band magnitude, RP-band flux, and BP-RP. This was to be expected as the colour ratio BP-RP has a clear linear dependence on the effective temperature. Cooler stars have their light peaks shifted towards the redder end of the spectrum and vice versa for hotter stars. Running the algorithm resulted in a MSE of 0.004250197448159388.



(a)



(b) Sharp increase in cross-validation score for the first 5 inputs, showing that most of the weight in the dataset is contained in top 5 features.

Figure 8

Training RFR came with its own advantages with respect to linear regression and MLPRegressor. It's ability to handle non-linear relationships amongst the observational parameters as well as flexibility to the GaiaDR3 by iterating over a hyperparameter grid, gave 100 trees, no maximum tree depth, 2 samples required to split a node and 1 sample required to be at a leaf node as the best performing hyperparameters. This however came with the limitation of a reduced testing set from 320k to 80k as a trade-off for the running time. This was due to averaging the results of 100 decision trees, with each trained on all 5 sub folds of the training data, thus the program was able to converge to a set of effective temperatures with an MSE of 0.0021473770210116453. The program returned right ascension, declination, error in declination, parallax, and error in G-band magnitude as the 5 most important features. In the figure below, the results are pictured.

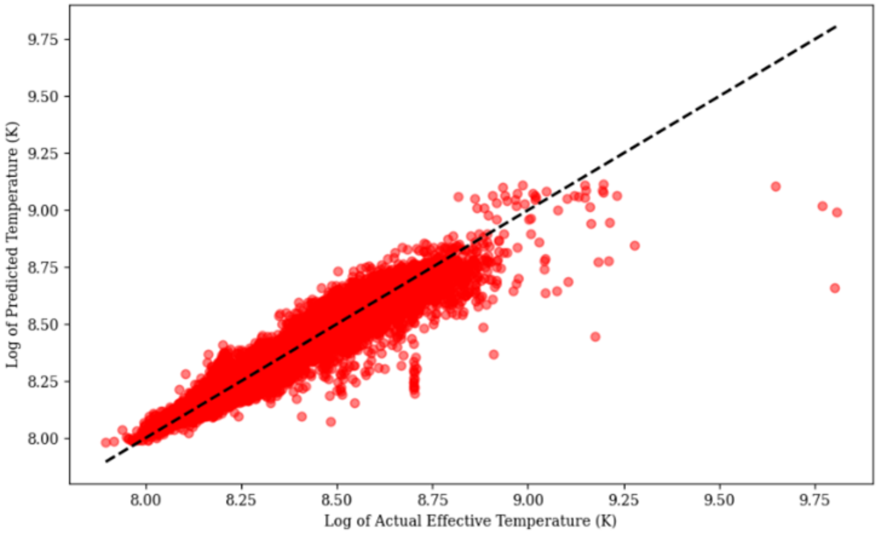


Figure 9

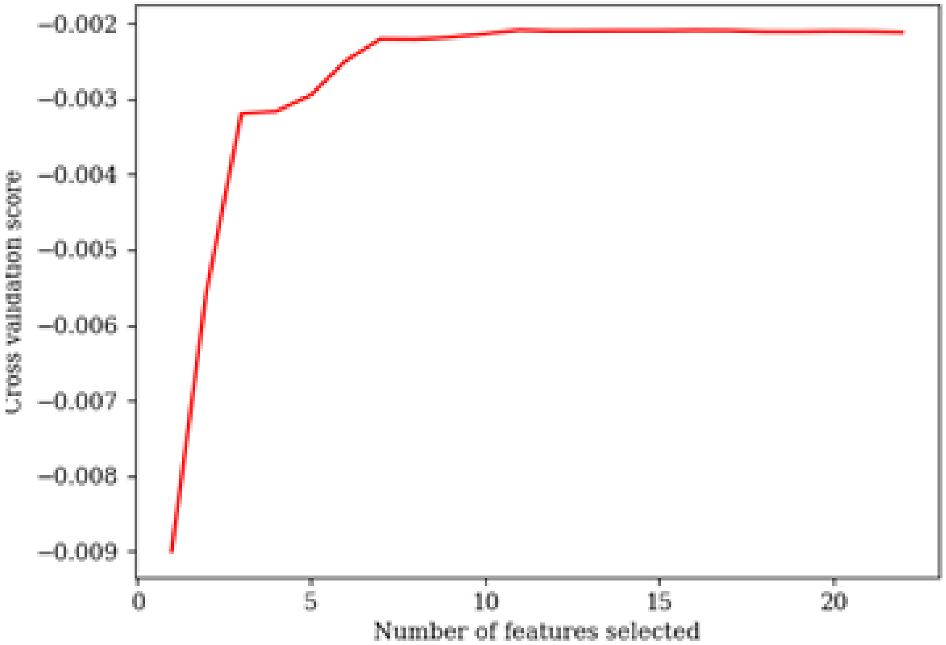


Figure 10: The first 6 features contained most of the variance in the data, with the peak cross-validation score at 10 features. The purpose of such a curve is to see how well the model generalises to unseen data for a given number of features. Note, that adding more features after 10 showed diminishing returns and thus was redundant to the performance of the model. This situation I believed to be a sign of overfitting, possibility to due to trees being allowed an unlimited depth and thus when the model was applied to the full testing set only the 10 most important features were used.

MLPRegressor was the most flexible out of the three algorithms but could not be used as an estimator for RFE leading me to use permutation importance. This estimated the importance of features by permuting the values of each feature and measuring the decrease in the model's performance. Trying different combinations of features alongside grid search cross-validation was very computationally intensive to find the best performing hyperparameters, so a reduced testing set was also used. The MSE returned was 0.002960788113609551 and the selected hyperparameters were the tanh activation function, alpha=0.01, 23 neurons in one hidden layer, invscaling learning rate, and ‘adam’ as the solver. The tanh activation function outputs values from -1 to 1 confirming the non-linear relationships in the data. Alpha=0.01 suggested that the model used a moderate amount of regularisation to prevent overfitting, indicating a good balance between under and overfitting, which boded well for full scale testing. 23 neurons were opted for one hidden layer so the model would not be so complex to risk overfitting. Invscaling meant that the learning rate decreased as the training progressed, which allowed the model to make larger updates in earlier stages and smaller, more precise updates later which may have helped the model converge more effectively. The permutation importance returned BP-band magnitude, BP-band flux, G-band magnitude, RP-band magnitude, BP-RP.

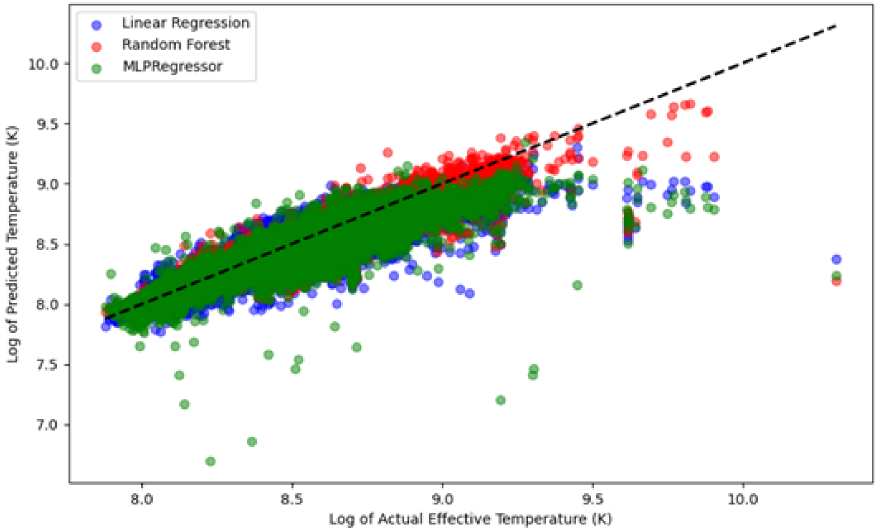


Figure 11

In the figure above all three algorithms were applied to the full testing set after being trained on the same data (random state kept constant). The results were, Linear Regression MSE: 0.0036466340975548892 (0.28 seconds running time), Random Forest Regressor MSE: 0.0017899202586290025 (approximately 4 minutes running time), MLPRegressor MSE: 0.002507264324539177 (12 seconds running time). All three algorithms showed scattering at higher temperatures as well as predominantly lower predicted temperatures overall. I believed this to be due to interstellar extinction, this scattering and absorption of an object's emitted light could have made it appear dimmer than it is. The amount of reddening the object experienced depended on distance from the galactic centre, the density of the dust and gas, with the effect being more pronounced in the BP-band, where the stars are brighter and more susceptible to extinction. Additionally, systematic errors (such sensor noise or calibration error) within the Gaia photometric instruments used to record the flux measurements could have compounded at higher effective temperatures where the equipment may not have been as sensitive, this is shown by the plot of effective temperature against error in FBP. This could lead to ML models trained on this data to consistently under predict effective temperatures.

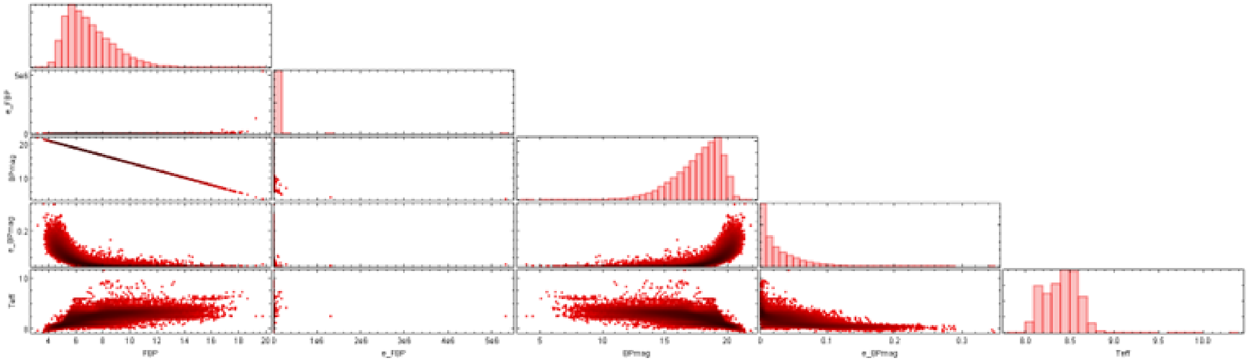


Figure 12: Corner plot of the blue photometric band against the effective temperature. Note that for the higher effective temperatures there is a lot of scatter in flux detected, magnitude and error in flux detected.

Although applying PCA to aid in predicting effective temperatures may have given the algorithms a boost in performance as well reduced running time by selecting the first few components, I opted to keep inputs interpretable and feed correlated inputs into the algorithms (as pictured in the main corner plot) aid me in understanding the behaviours of the results. To conclude on the best performing algorithm, I used a weighted sum of the MSE and running time 'Bang for Buck' to reflect the importance of each metric. Assigning 0.7 to the MSE (since it's more critical for model accuracy) and a 0.3 to the running time (since computational efficiency is also important but less critical than accuracy). Linear Regression = 0.002608592135117441, Random Forest Regressor = 0.0013099603793145015 and MLPRegressor =

0.001986192962695786. Random Forest Regressor therefore offered the best balance between accuracy and computational efficiency among the three algorithms.

3 X-ray Binary Classification

3.1 Context

HMXBs, LMXBs, and CVs are distinguished by their unique X-ray emissions and the specifics of their component stars. HMXBs are characterized by a massive normal star, which dominates the emission of optical light, and a compact object, such as a neutron star or black hole, which is the primary source of X-rays [40]. The variability of HMXBs is manifested as X-ray pulsars, a result of matter being magnetically funneled into the poles of the compact companion during accretion. This process is crucial for the emission of X-rays, making HMXBs one of the most renowned types of X-ray binaries. LMXBs are identified by having one component that is either a black hole or neutron star, with the other component being a donor star. This donor star fills its Roche lobe and transfers mass to the compact object, leading to the emission of almost all radiation in X-rays [41]. LMXBs are among the brightest objects in the X-ray sky but are relatively faint in visible light. Their variability is primarily observed in X-ray bursters, which are thermonuclear explosions triggered by the accretion of Hydrogen and Helium. CVs, consist of a white dwarf and a normal star companion. These systems are typically small, with orbital periods ranging from 1 to 10 hours. The normal star loses material onto the white dwarf through accretion, and due to the white dwarf's high density, a significant amount of gravitational potential energy is converted into X-rays during this process [42]. As a result, CVs are faint in X-rays, placing them just above the coronal X-ray sources and far below the X-ray binaries in terms of the intensity of their X-ray emissions.

3.1.1 Class 0

By training on both x-ray binaries and 'uninteresting' objects such as low and high mass stars, young stellar objects and active galactic nuclei, the model learnt to differentiate between class 0 and the characteristics that define x-ray binaries. This balanced approach helps prevent the model from becoming biased toward x-ray binaries, which could lead to overfitting to the outlier characteristics and poor generalisation to new data. When a model is only trained on outliers, every slight deviation from the norm can be misinterpreted as an outlier. Including class 0 objects helps the model recognise and tolerate normal variations within photometric and x-ray measurements, thereby reducing false positives [43]. For most astronomical datasets, class 0 objects represent most of the data. Excluding these would provide a skewed representation of reality, leading the model to learn from an unrepresentative sample. This can degrade performance when the model is applied to a full testing set where class 0 objects are predominant.

3.1.2 Training Set Proportions

I prepared a training set consisting of 5000 class 0 objects, 300 CVs, 20 HMXBs, and 32 LMXBs. CVs are generally more common than HMXBs and LMXBs as they involve relatively low-mass main-sequence stars or slightly evolved stars, which are more abundant in the universe because the initial mass function favours the formation of lower-mass stars [44]. HMXBs and LMXBs involve either neutron stars or black holes, resulting from the evolution of more massive stars, which are significantly rarer. Additionally, LMXBs are more common than HMXBs because low-mass stars have longer lifespans, allowing more time for the binary system to evolve into an LMXB, and because these systems are generally more stable during mass transfer, increasing their chances of detection and survival over time. The formation pathway for CVs is somewhat more straightforward and has fewer catastrophic events that could disrupt the binary system. While CVs still require close binary systems and common envelope phases for the white dwarf to begin accreting from its companion, these events are less disruptive compared to the supernova explosions needed to form compact objects in HMXBs and LMXBs.

3.2 Preprocessing

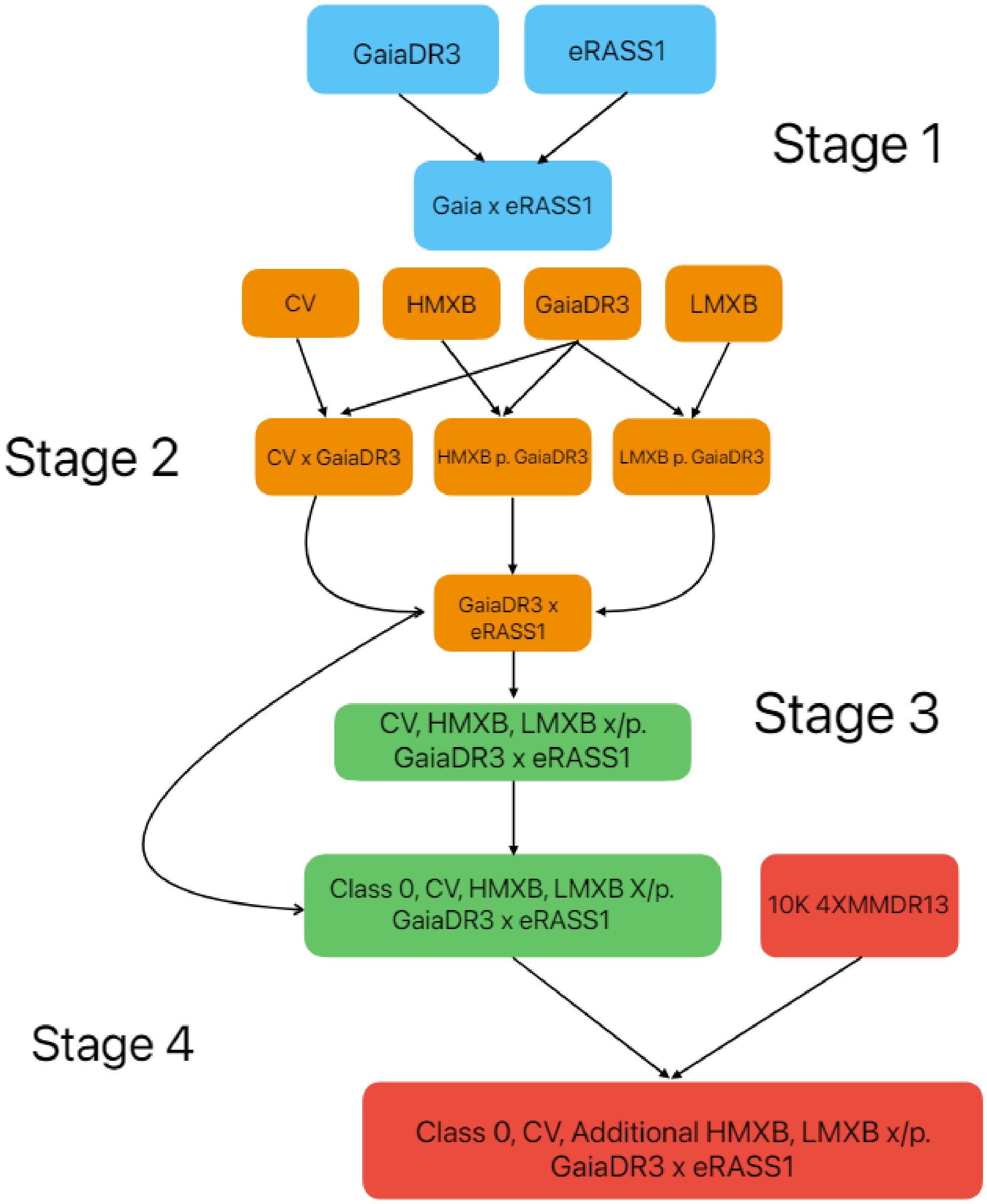


Figure 13

Note that in ‘stage 1’, the two catalogues were cross matched to 5 arcseconds on CDS X-Match. As the full GaiaDR3 catalogue contained approximately 1.8 billion sources, it is crucial to understand that this meant per eRASS1 source I assumed many duplicate GaiaDR3 sources in the 5 arcsecond window. This skewed the crossmatched catalogue in favour of completeness than purity when searching for x-ray binary outliers, of which this manifested itself later when looking at the classification reports in the training and testing stage. In stage 2, cross matching GaiaDR3 with the known CV catalogue introduced more duplicates into this task due to the sheer number of Gaia sources compared to CVs. Pair-matching the known HMXB, LMXB catalogues with GaiaDR3 through the shared ‘GaiaID’ column proved simple. At the end of stage 2 came the first training set by concatenating relevant photometric, x-ray, parallax, proper motion, positional columns on TOPCAT. 5000 class 0 zero sources were taken from the GaiaDR3 x eRASS1 catalogue. I was able to assume within reason these were all true class 0 objects by only sampling those with parallax over error greater than 5, the high precision in parallax measurements increasing statistical confidence in the properties derived from these measurements [45]. To avoid subset bias in sampling class 0 objects all from the GaiaDR3 x eRASS1 data, the random index was utilised to take a random sample. This confidence is crucial for classifying these objects as typical stars rather than x-ray binary phenomena I am seeking. To provide enough known sources of HMXBs and LMXBs such that the algorithm would have enough variation in x-ray binary to learn from, extra sources were pair-matched from the 10K 4XMM-DR13 catalogue and concatenated to form the final training set. The photometric bands, parallax, proper motion etc were the same as discussed in the regression task, however the source x-ray fluxes came in form of 9 bands ranging from soft (0.2-0.5 keV) to medium (2.0-5.0 keV) to hard (7.2-8.2 keV).

3.3 Results

The Logistic Regression model was internally split (20:80) in favour of testing. To find the optimal combination of hyperparameters and the features that carry the most weight, I opted for robust scaling for all three algorithms unlike standard scaling, which used the mean and standard deviation and could be significantly affected by outliers whereas robust scaling used the median and IQR, making it resilient to outliers. Stratified samples were also used for all three algorithms, such it was able to learn from representative proportions of each class. A random state was specified to compare the performance of all three models on the same shuffle of the data. For multinominal classification in imbalanced datasets, accuracy scoring was replaced with the weighted F1 score in all three algorithms, as insight into the harmonic mean between completeness (recall) and purity (precision) of classification is far more important when training for outlier detection. The best parameters returned were: C = 1, balanced class weight, true fitted intercept, L1 penalty, liblinear solver, and 0.01 stopping tolerance. Using RFE, the 5 most important features were parallax, parallax over error, G-band mean flux over error, G-band magnitude, and BP-RP. Note that Logistic Regression failed to give any importance to the source x-ray flux measurements, which I believe to be a crucial reason for the poor generalisation to the full-scale testing later. The model performed exceptionally well on class 0, with high precision, recall, and F1-score. However, it struggled with class 1, 2, and 3, which had lower precision, recall, and F1-score. At the very least, I had some degree of confidence in the model's ability to distinguish uninteresting objects. The macro average precision, recall, and F1-score indicated that the model performed moderately well across all classes, but with a significant disparity between class 0 and 1,2,3. The weighted average precision, recall, and f1-score suggested that the model performed exceptionally well on the majority class but still struggled with the minority classes. The figures below illustrate these results.

Table 1: Logistic Regression Classification Report

Class	**Precision**	**Recall**	**F1-score**	**Support**
0	0.99	0.98	0.99	3856
1	0.78	0.86	0.82	238
2	0.33	0.38	0.35	16
3	0.43	0.24	0.31	25
Accuracy		0.97 (4135)		
Macro Avg	0.63	0.61	0.62	4135
Weighted Avg	0.97	0.97	0.97	4135

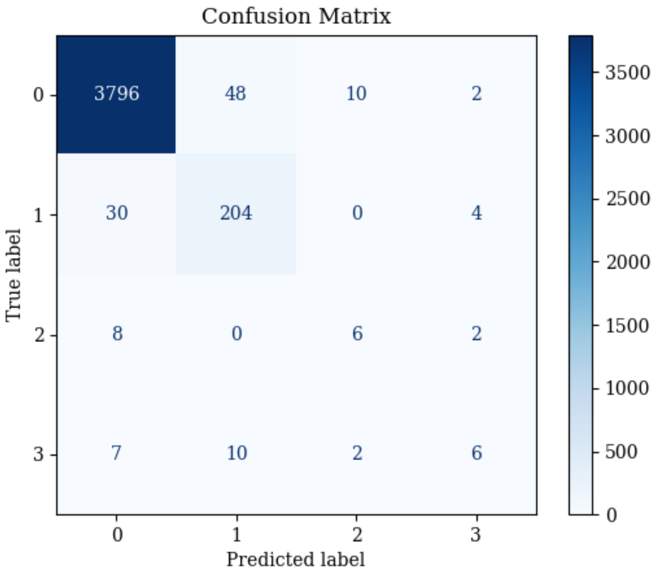


Figure 14

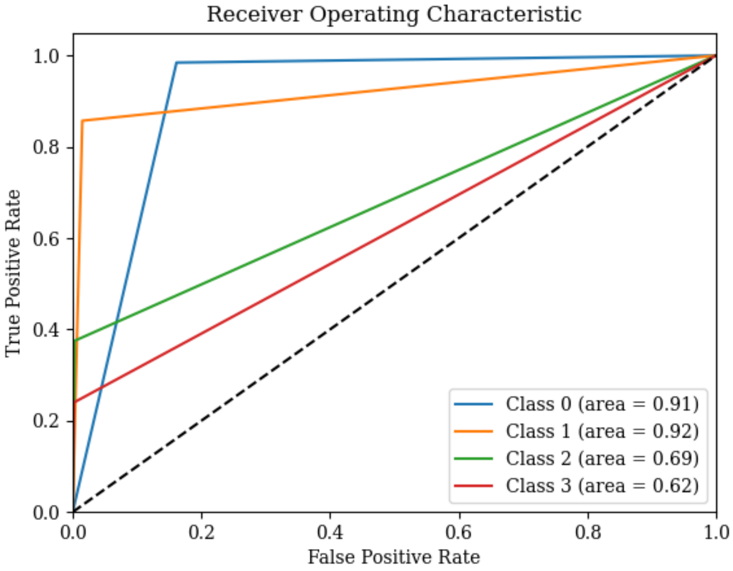


Figure 15

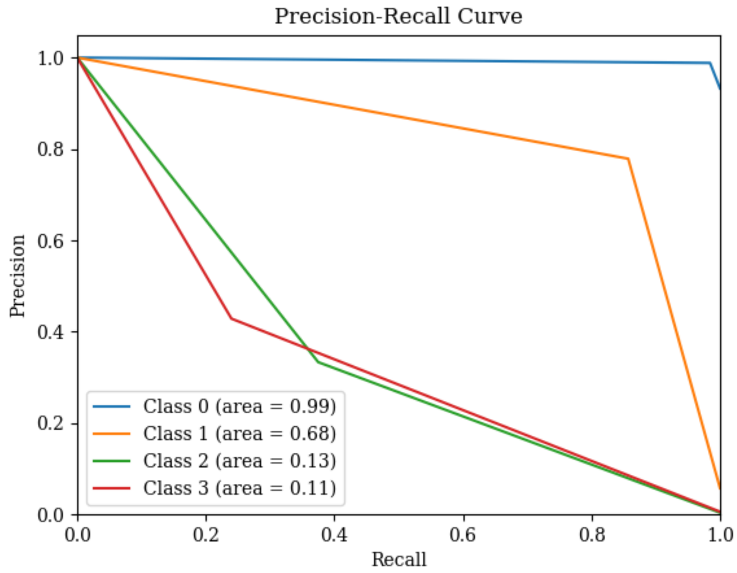


Figure 16

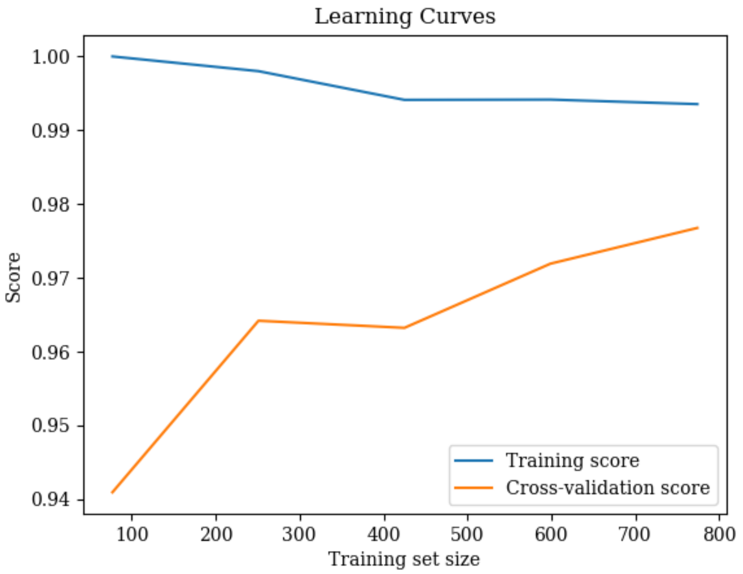


Figure 17

Table 2: Logsitic Regression Classification Report with PCA

Class	Precision	Recall	F1-score	Support
0	0.99	0.98	0.99	3856
1	0.76	0.85	0.80	238
2	0.29	0.25	0.27	16
3	0.30	0.32	0.31	25
Accuracy		0.97 (4135)		
Macro Avg	0.58	0.60	0.59	4135
Weighted Avg	0.97	0.97	0.97	4135

Table 3: PCA Components and Validation Score

Metric	Value
Best Number of PCA Components	34
Validation Score	0.9675937122128174

The application of PCA did not improve the model's performance for all the classes. In fact, it negatively impacted the model's ability to correctly classify the minority classes (classes 1, 2, and 3), as indicated by the decreased precision, recall, and F1-score for these classes. The model's performance on class 0 remained high, but the overall performance has been slightly affected by the PCA transformation. This suggested that while PCA could have been useful for dimensionality reduction and noise reduction, yet not beneficial when paired with Logistic Regression. When applying logistic regression to the full 400K testing set along with RFR and MLPClassifier, no random state was specified to trial the model on 10 random shuffles. In doing so, 'good candidates' were found by taking the modal class allocation per source out of the 10 runs of the program. The results are listed below are for Logistic Regression.

Table 4: Number of Classes Identified

Class	**Count**
0	267611
1	99611
2	6239
3	22008

RFR was prepared for full scale testing in the same way as logistic regression. The optimal hyperparameters returned were, ‘True’ bootstrap, the model benefited from creating more diverse trees to capture complex patterns in the data. A tree depth of 20 suggested that the model captured complex relationships without overfitting too much. One sample needed to form a leaf meant a far more intricate tree structure, capturing more nuanced patterns in the data. The risk of overfitting was lowered by 10 samples needed to split the node, ensuring that model didn’t split without many samples being drawn from. The number of trees remained at the default value, suggesting a balance already existed in the model between performance and computational efficiency. The 5 features the carried the most importance through RFE were: ’MLFluxP4’, ’RPlx’, ’RFG’, ’Gmag error’, ’RFRP’. Interestingly, with the increased flexibility of RFR compared with logistic regression placed more importance was placed on the noise surrounding the G-band inputs as well as the medium x-ray band, indicating that a more complex model was necessary to pick up on the importance of the source x-ray fluxes when finding x-ray binary outliers. The figures below illustrate these results.

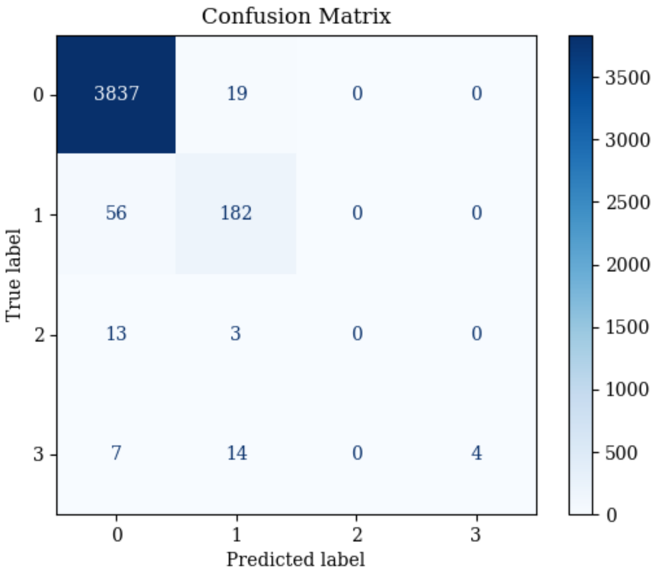


Figure 18

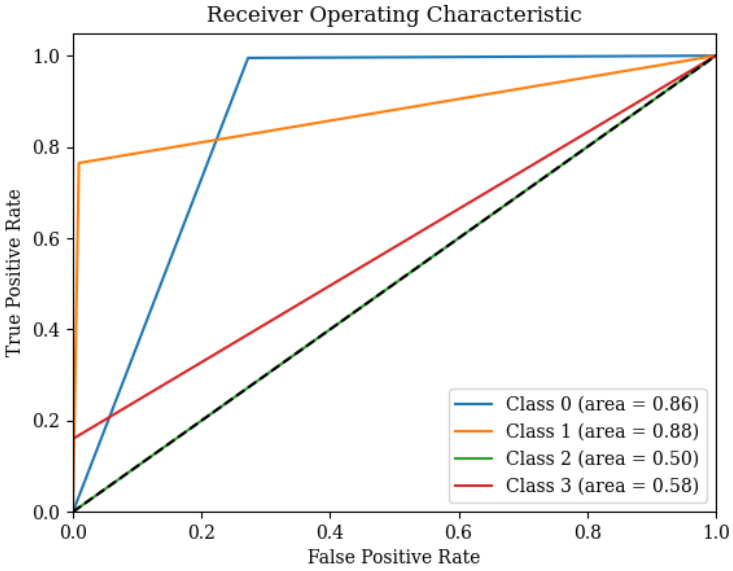


Figure 19

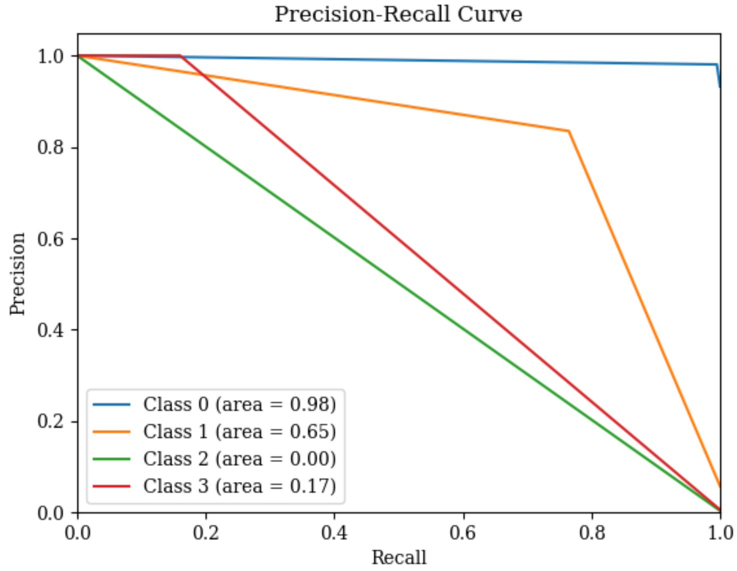


Figure 20

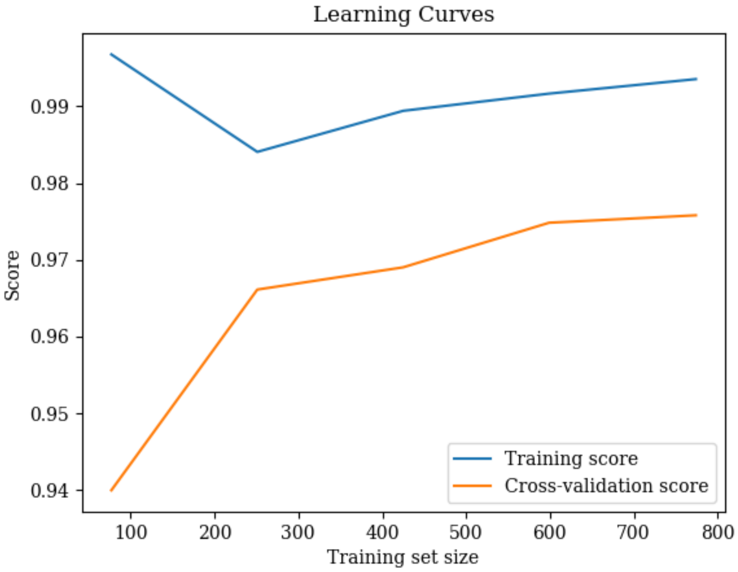


Figure 21

Table 5: Random Forest Classification Report

Class	**Precision**	**Recall**	**F1-score**	**Support**
0	0.98	1.00	0.99	3856
1	0.83	0.76	0.80	238
2	0.00	0.00	0.00	16
3	1.00	0.16	0.28	25
Accuracy		0.97 (4135)		
Macro Avg	0.70	0.48	0.52	4135
Weighted Avg	0.97	0.97	0.97	4135

The precision, recall, and F1-score for class 2 were all 0 in the original model without PCA, indicating that the model was not able to correctly classify any instances of class 2. In the model with PCA, the precision and recall for class 2 were both 0, but the F1-score was 0.48, suggesting that the model did not perform well on class 2, but slightly better than the original model. With class 3, the original model and PCA were able to classify all instances, but the F1-score was 0.48 in the PCA model, which suggested that while the model still performed well on class 3 instances, the overall performance had slightly decreased. The accuracy and macro average scores were the same in both models, indicating no significant change to performance with PCA. Weighted average scores were also the same in both models, further supporting the idea that the overall performance has not changed significantly with PCA. Overall, applying PCA with the optimal hyperparameters for RFR to seek a performance boost did not significantly improve the model's performance on the training data.

Table 6: Random Forest Classification Report with PCA

Class	Precision	Recall	F1-score	Support
0	0.98	0.99	0.99	3856
1	0.85	0.74	0.79	238
2	0.00	0.00	0.00	16
3	1.00	0.32	0.48	25
Accuracy		0.97 (4135)		
Macro Avg	0.71	0.51	0.56	4135
Weighted Avg	0.97	0.97	0.97	4135

Table 7: PCA Components and Validation Score

Metric	Value
Best Number of PCA Components	13
Validation Score	0.9717049576783555

Table 8: Number of Classes Identified

Class	Count
1	279229
0	116105
3	135

The best performing set of hyperparameters were ‘tanh’ as the activation, alpha = 0.01, 42 neurons in one hidden layer, constant learning rate and ‘lbfgs’ as the solver. The tanh function outputs values between -1 and 1, which helped in learning complex patterns that are not linearly separable. A small value of alpha indicated that the model actively used regularisation to prevent overfitting. 42 neurons in one layer were selected to match the number of inputs in the data instead of the default 100 arbitrary neurons, with one layer usually being enough to provide a good fit. A constant learning rate meant an equal update to the weights per iteration of the program which led to stable convergence, as the learning rate did not need to change to mitigate underfitting over time. The five parameters returned by permutation importance (as in regression) were: MLFluxP3, error in RPmag, MLFluxP4, MLFluxP1, error in Gmag. Soft to medium source x-ray fluxes proved most detrimental to MLPClassifier's performance to leave out along with errors in RP-band magnitude and G-band magnitude compared with the positional and parallax data. Learning from data on soft to medium x-ray flux bands had the potential to differentiate between CVs, LMXBs, and HMXBs based on accretion processes and spectral characteristics indicative of the temperature and density of accreting material. Errors in the G-band and RP-band proved significant as they could have affected the accuracy of optical counterparts; high errors suggesting variability or unusual features important for classification. To note, As the data is heavily imbalanced, these features may have been valuable to the model for distinguishing rarer classes, thus ranked as more important.

Table 9: MLP Classification Report

Class	**Precision**	**Recall**	**F1-score**	**Support**
0	0.99	0.99	0.99	3856
1	0.83	0.83	0.83	238
2	0.29	0.12	0.17	16
3	0.57	0.32	0.41	25
Accuracy		0.97 (4135)		
Macro Avg	0.67	0.57	0.60	4135
Weighted Avg	0.97	0.97	0.97	4135

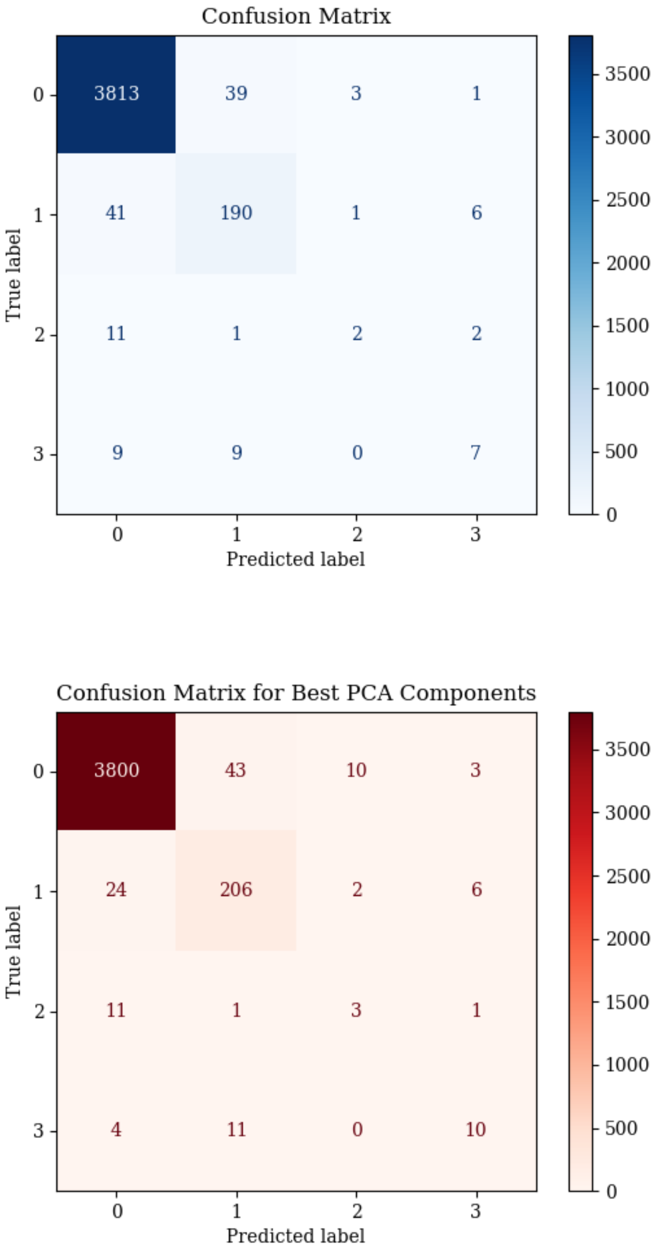
Table 10: MLP Classification Report with PCA

Class	Precision	Recall	F1-score	Support
0	0.99	0.99	0.99	3856
1	0.79	0.87	0.83	238
2	0.20	0.19	0.19	16
3	0.50	0.40	0.44	25
Accuracy		0.97 (4135)		
Macro Avg	0.62	0.61	0.61	4135
Weighted Avg	0.97	0.97	0.97	4135

Table 11: PCA Components and Validation Score

Metric	Value
Best Number of PCA Components	29
Validation Score	0.9746070133010882

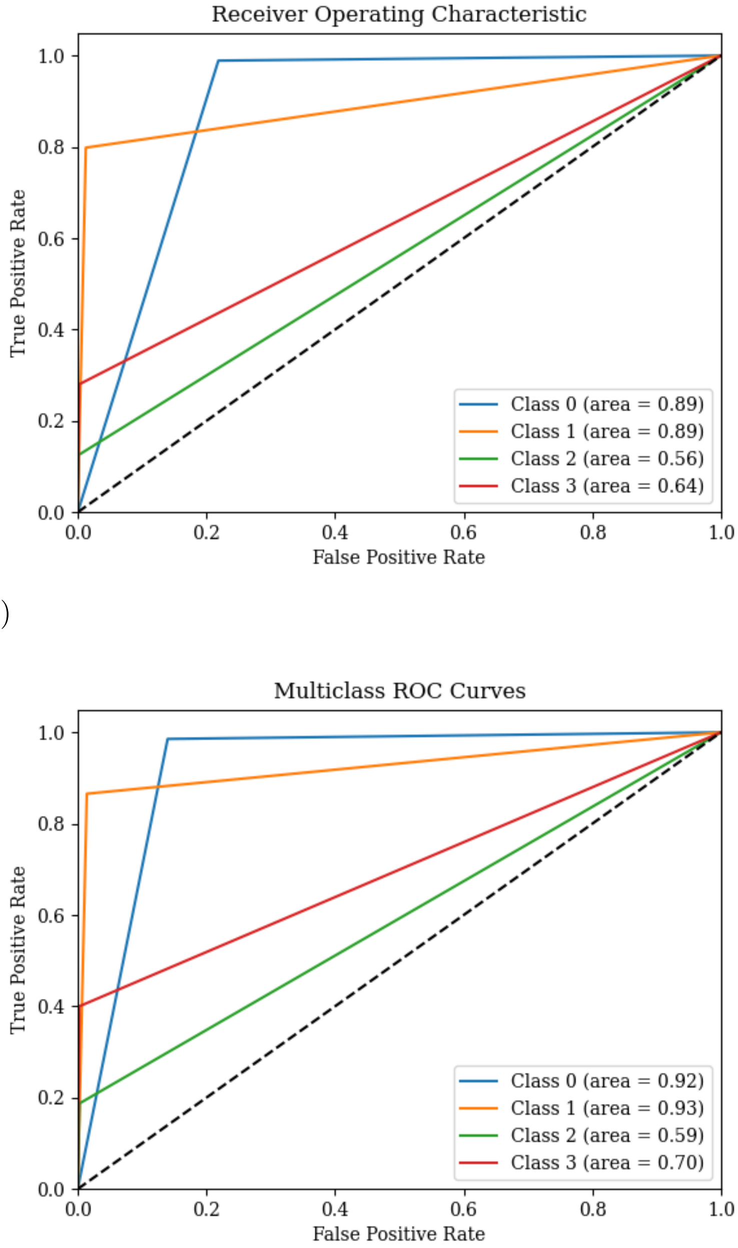
Comparing the two classification reports, on the same imbalanced internal split, it was evident that both models performed similarly in terms of overall accuracy (0.97), yet differed in handling specific classes. The first model achieved a very high precision, recall, and f1-score for Class 0, moderate effectiveness for Class 1 with balanced precision and recall, both at 0.83, but struggled with Class 2 and Class 3. The PCA model showed a nuanced shift, slightly reducing precision but improving recall for class 1 and class 3. This resulted in better f1-scores, indicating an improved balance between identifying true positives and avoiding false negatives. The slight improvements in macro-average recall with the PCA model indicated greater sensitivity to minority classes despite a minor drop in precision. This shift suggested that PCA helped in reducing overfitting and potentially would allow the model to generalise better over a reduced feature space during full scale testing. Thus, the choice between these models depended on whether the goal was completeness in detecting x-ray binaries, maintaining purity across x-ray binary predictions, or keeping input features interpretable when full scale testing the model. The figures below show the results of this comparison.


Confusion Matrix
True label
Predicted label
0
1
2
3
3813
39
3
1
41
190
1
6
11
1
2
2
9
9
0
7
0
500
1000
1500
2000
2500
3000
3500
Confusion Matrix for Best PCA Components
3800
43
10
3
24
206
2
6
11
1
3
1
4
11
0
10

(a)

(b)

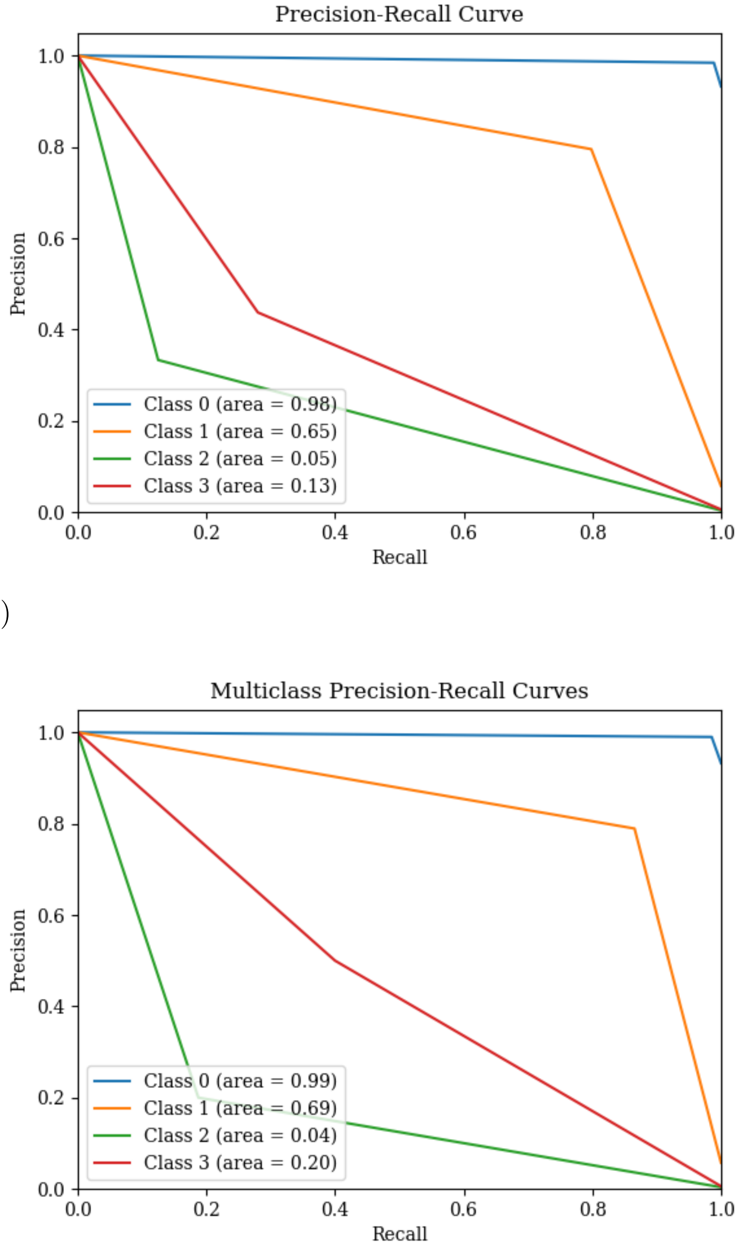
Figure 22


Receiver Operating Characteristic
1.0
0.8
0.6
0.4
0.2
0.0
True Positive Rate
0.0
0.2
0.4
0.6
0.8
1.0
False Positive Rate
Class 0 (area = 0.89)
Class 1 (area = 0.89)
Class 2 (area = 0.56)
Class 3 (area = 0.64)
Multiclass ROC Curves
1.0
0.8
0.6
0.4
0.2
0.0
True Positive Rate
0.0
0.2
0.4
0.6
0.8
1.0
False Positive Rate
Class 0 (area = 0.92)
Class 1 (area = 0.93)
Class 2 (area = 0.59)
Class 3 (area = 0.70)

(a)

(b)

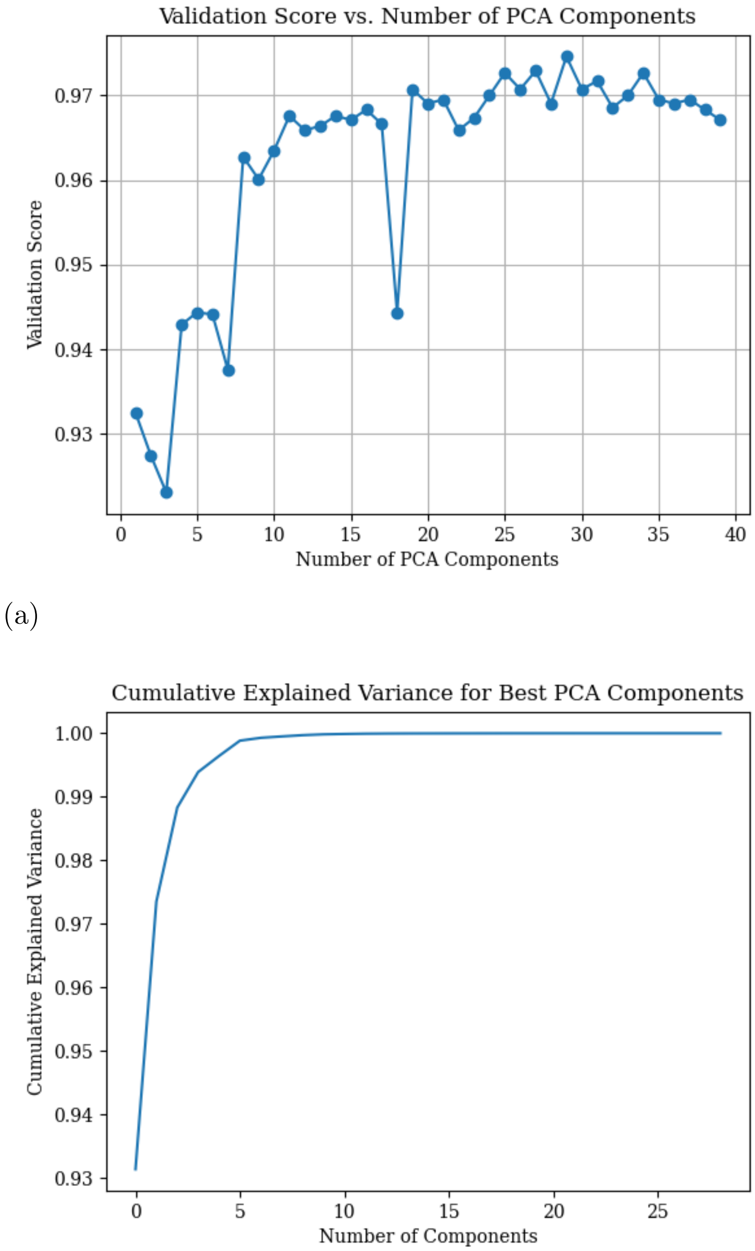
Figure 23


Precision-Recall Curve
Precision
1.0
0.8
0.6
0.4
0.2
0.0
0.0
0.2
0.4
0.6
0.8
1.0
Recall
Class 0 (area = 0.98)
Class 1 (area = 0.65)
Class 2 (area = 0.05)
Class 3 (area = 0.13)
Multiclass Precision-Recall Curves
Precision
1.0
0.8
0.6
0.4
0.2
0.0
0.0
0.2
0.4
0.6
0.8
1.0
Recall
Class 0 (area = 0.99)
Class 1 (area = 0.69)
Class 2 (area = 0.04)
Class 3 (area = 0.20)

(a)

(b)

Figure 24


Validation Score vs. Number of PCA Components
Validation Score
0.97
0.96
0.95
0.94
0.93
0
5
10
15
20
25
30
35
40
Number of PCA Components
Cumulative Explained Variance for Best PCA Components
Cumulative Explained Variance
1.00
0.99
0.98
0.97
0.96
0.95
0.94
0.93
0
5
10
15
20
25
Number of Components

(a)

Figure 25

4 Concluding Remarks

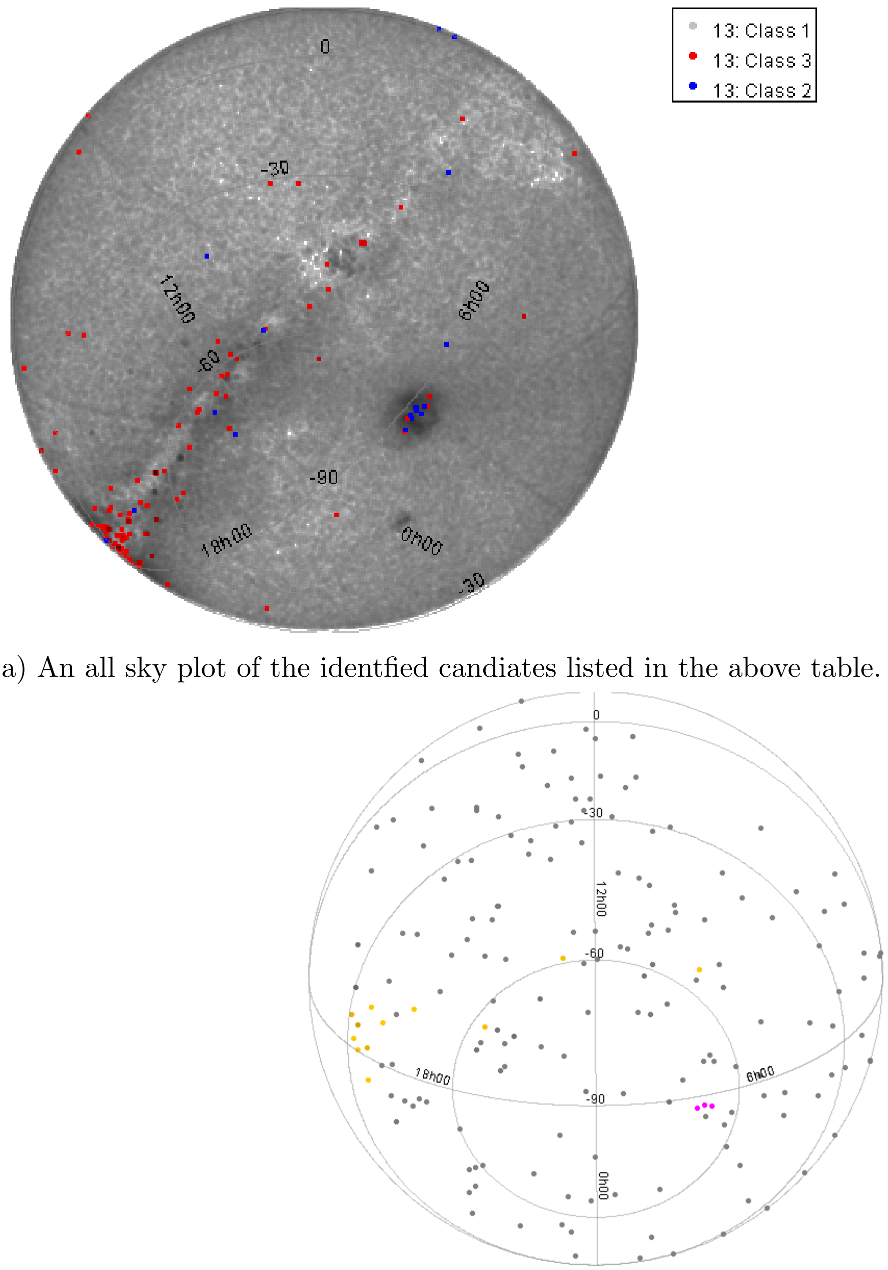
Considering the candidates for each respective class identified by each algorithm, the final classification count below was found by pair matching common GaiaIDs between RFR and MLPClassifier. Logistic Regression overall proved too coarse in its classification, predicting far too many HMXBs and LMXBs than would be reasonable for comparison due to a lack of flexibility in the model. Thus, I made the decision to not include these classifications when filtering for a final set.

Limitations for the processes used in classification were initial cross matches between GaiaDR3 and eRASS1 were made to within 5 arcseconds with all matches retained rather than just the 'best' to ensure completeness in the data, thus it was sensible to assume many duplicate GaiaDR3 sources per eRASS1 source. Whilst this may have included nearly all true positives for classes 1, 2 and 3, this brought a lot of false positive impurities into the training of ML models leading to class confusion in testing. My initial proportions of class 0, 1, 2 and 3 in the training set may not have reflected the true proportions in the GaiaDR3 x eRASS1 cross match, and there might have not been enough variance in class 0 objects in training to recognise the dominance of 'true' class 0 in testing. RFR and MLPClassifier had a lot of bias towards the classifying class 1 objects, and hyperparameters that were a 'good' fit for the training data could have not generalised well to the testing set, and these algorithms could have been trained on class 0 objects that may still have been 'interesting' sources such as x-ray emitting main sequence binaries and YSOs (which were mistakenly taken to be a class 0) showing eruptions like CVs.

To extend the processes used in classification, I would construct new parameters taking the magnitude of a combination of consecutive source x-ray fluxes in the GaiaDR3 x eRASS1 catalogue as well as total optical flux over total x-ray source flux, ranging from soft all the way to hard to see if these parameters improve the performance and speed of convergence of the classifiers by mitigating against feeding the classifiers 0 values for certain x-ray source flux bands and well as providing fewer inputs. Alongside supplementing more 'true' class 0 objects in the training set, I would investigate the calibration of aperture corrected net x-ray fluxes in the Chandra source catalogue and mean x-ray fluxes from XMM-Newton Serendipitous source catalogue to provide more known examples of HMXBs, LMXBs and CVs to train on before again trialling on the GaiaDR3 x eRASS1 catalogue to look for new x-ray binary candidates.

Table 12: Number of Classes Identified

Class	**Count**
0	117420
1	277828
2	18
3	203



(a) An all sky plot of the identfied candiates listed in the above table.

Figure 26: A 5 arcsecond cross match between SIMBAD and my identified candidates to confirm if some of the canddiates already exist. Grey points in this instance are class 1, pink is class 3 and yellow is class 2.

5 Bibliography

1. Zhang Y, Zhao Y. Astronomy in the big data era. Data Science Journal [Internet]. 2015 May 22 [cited 2024 Apr 15];14(0):6. Available from: https://datascience.codata.org/articles/10.5334/dsj-2015-011

2. Brunner R. Massive datasets in astronomy [Internet]. California Institute of Technology; 2 p. Available from: https://www.cs.princeton.edu/courses/archive/spring04/c

3. Xu Y, Liu X, Cao X, Liu E. Artificial intelligence: A powerful paradigm for scientific research. The Innovation [Internet]. 2021 Nov 28 [cited 2024 Apr 13];2(4):1. Available from: https://www.sciencedirect.com/science/article/pii/S2666675821001041

4. What is machine learning (Ml)? — ibm [Internet]. [cited 2024 Apr 2]. Available from: https://www.ibm.com/topics/machine-learning

5. Regression in machine learning: what it is examples — built in [Internet]. [cited 2024 Apr 1]. Available from: https://builtin.com/data-science/regression-machine-learning

6. Brownlee J. Regression Metrics for Machine Learning.

7. Chapter 7: correlation and simple linear regression — natural resources biometrics [Internet]. [cited 2024 Apr 5]. Available from: https://courses.lumenlearning.com/suny-natural-resources-biometrics/chapter/chapter-7-correlation-and-simple-linear-regression/

8. Bootstrap aggregation, random forests and boosted trees — quantstart [Internet]. [cited 2024 Apr 4]. Available from: https://www.quantstart.com/articles/bootstrap-aggregation-random-forests-and-boosted-trees/

9. Decision trees - an overview — sciencedirect topics [Internet]. [cited 2024 Apr 7]. Available from: https://www.sciencedirect.com/topics/computer-science/decision-trees

10. Sachinsoni. Mastering random forests: unraveling the magic of ensemble learning [Internet]. Medium. 2023 [cited 2024 Apr 1]. Available from: https://medium.com/@sachinsoni600517/mastering-random-forests-unraveling-the-magic-of-ensemble-learning-e80472723cee

11. Greenwell BB B. Chapter 9 decision trees — hands-on machine learning with r [Internet]. [cited 2024 Feb 17]. Available from: https://bradleyboehmke.github.io/HOML

12. Khan M. Automated Prediction of Good Dictionary EXamples (GDEX): A Comprehensive Experiment with Distant Supervision, Machine Learning, and Word Embedding-Based Deep Learning Techniques. :7. Available from: https://www.researchgate.net/publication/354354484$_{A}utomated_{P}rediction_{o}f_{G}ood_{D}ictionary_{E}Xar$
$Based_{D}eep_{L}earning_{T}echniques/download$

13. Shah R. Tune hyperparameters with gridsearchcv [Internet]. Analytics Vidhya. 2021 [cited 2023 Nov 22]. Available from: https://www.analyticsvidhya.com/blog/2021/06
hyperparameters-with-gridsearchcv/

14. scikit-learn [Internet]. [cited 2023 Oct 29]. Sklearn. Neural$_{n}etwork.Mlpregressor.Availablef$
$https : //scikit-learn/stable/modules/generated/sklearn.neural_{n}etwork.MLPRegressor.html1$

16. scikit-learn [Internet]. [cited 2024 Apr 23]. 1.17. Neural network models (Supervised). Available from: https://scikit-learn/stable/modules/neural$_{n}etworks_{s}upervised.h$

17. Otten NV. Spot Intelligence. 2024 [cited 2024 Jan 15]. Learning rate in machine learning and deep learning made simple. Available from: https://spotintelligence.com/2024/02/19/learning-rate-machine-learning/

18. Kotsiantis SB, Zaharakis ID, Pintelas PE. Machine learning: a review of classification and combining techniques. Artif Intell Rev [Internet]. 2006 Nov 1 [cited 2024 Apr 20];26(3):159–90. Available from: https://doi.org/10.1007/s10462-007-9052-3 19. Heydarian M, Doyle TE, Samavi R. Mlcm: multi-label confusion matrix. IEEE Access [Internet]. 2022 [cited 2024 Apr 23];10:19083–95. Available from: https://ieeexplore.ieee.org/abstract/document/9711932/

20. Pepe MS. Receiver operating characteristic methodology. Journal of the American Statistical Association [Internet]. 2000 Mar [cited 2024 Apr 23];95(449):308–11. Available from: http://www.tandfonline.com/doi/abs/10.1080/01621459.2000.

21. Saito T, Rehmsmeier M. The precision-recall plot is more informative than the roc plot when evaluating binary classifiers on imbalanced datasets. PLoS One [Internet]. 2015 Mar 4 [cited 2024 Apr 23];10(3):e0118432. Available from: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4349800/

22. Perlich C. Learning curves in machine learning. In: Sammut C, Webb GI, editors. Encyclopedia of Machine Learning [Internet]. Boston, MA: Springer US; 2010 [cited 2024 Apr 23]. p. 577–80. Available from: https://doi.org/10.1007/978-0-387-30164-8$_4$52

23. Ranganathan P, Pramesh CS, Aggarwal R. Common pitfalls in statistical analysis: Logistic regression. Perspect Clin Res [Internet]. 2017 [cited 2024 Apr 23];8(3):148–51. Available from: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC55437

24. Schonlau M, Zou RY. The random forest algorithm for statistical learning. The Stata Journal [Internet]. 2020 Mar [cited 2024 Apr 23];20(1):3–29. Available from: http://journals.sagepub.com/doi/10.1177/1536867X20909688

25. scikit-learn [Internet]. [cited 2024 Apr 12]. Sklearn. Neural$_network.Mlpclassifier.Available$
$https://scikit-learn/stable/modules/generated/sklearn.neural_network.MLPClassifier.html$

26. Khalid S, Khalil T, Nasreen S. A survey of feature selection and feature extraction techniques in machine learning. In: 2014 Science and Information Conference [Internet]. 2014 [cited 2024 Apr 23]. p. 372–8. Available from: https://ieeexplore.ieee.org/abstract/document/6918213?casa$_token =$
$NgaFzOvmDFMAAAAA : 3ASgZ6H9I6hKk6IYYa8KKufiIzNsdxx0ChOei_oh7hGKc2ZPc7$

27. Jolliffe IT, Cadima J. Principal component analysis: a review and recent developments. Phil Trans R Soc A [Internet]. 2016 Apr 13 [cited 2024 Apr 23];374(2065):20150202. Available from: https://royalsocietypublishing.org/doi/10.1098/rsta.2

28. Huppenkothen D, Ntampaka M, Ho M, Fouesneau M, Nord B, Peek JEG, et al. Constructing impactful machine learning research for astronomy: best practices for researchers and reviewers [Internet]. arXiv; 2023 [cited 2024 Apr 23]. Available from: http://arxiv.org/abs/2310.12528

29. Gaia data release 3 overview - gaia - cosmos [Internet]. [cited 2024 Apr 10]. Available from: https://www.cosmos.esa.int/web/gaia/data-release-3

30. Erosita-de:datarelease1 [Internet]. [cited 2024 Apr 21]. Available from: https://erosita.mpe.mpg.de/dr1/

31. Lin Y, Yang H, Hare J, Volkov I, Kargaltsev O. Multiwavelength catalog of 10,000 4xmm-dr13 sources with known classifications. Res Notes AAS [Internet]. 2024 Mar 13 [cited 2024 Apr 23];8(3):74. Available from: http://arxiv.org/abs/2402.15684

32. Fortin F, Kalsi A, García F, Chaty S. A catalogue of low-mass X-ray binaries in the Galaxy: from the INTEGRAL to the Gaia era [Internet]. 2024 [cited 2024 Apr 23]. Available from: https://ui.adsabs.harvard.edu/abs/2024arXiv240111931F

33. Fortin F, García F, Simaz Bunzel A, Chaty S. A catalogue of high-mass X-ray binaries in the Galaxy: from the INTEGRAL to the Gaia era. Astronomy and Astrophysics [Internet]. 2023 Mar 1 [cited 2024 Apr 23];671:A149. Available from: https://ui.adsabs.harvard.edu/abs/2023AA...671A.149F

34. Ritter H, Kolb U. Catalogue of cataclysmic binaries, low-mass X-ray binaries and related objects (Seventh edition). Astronomy and Astrophysics [Internet]. 2003 Jun 1 [cited 2024 Apr 23];404:301–3. Available from: https://ui.adsabs.harvard.edu/abs/2003AA...404..301R

35. 11.3.3 General Stellar Parametrizer from Photometry (Gsp-phot) 11.3 Apsis modules Chapter 11 Astrophysical parameters Part III Gaia data analysis Gaia Data Release 3 Documentation release 1.3 [Internet]. [cited 2024 Apr 18]. Available from: https://gea.esac.esa.int/archive/documentation/GDR3/Data$_{a}nalysis/ch$

36. Minardi S, Harris RJ, Labadie L. Astrophotonics: astronomy and modern optics. Astron Astrophys Rev [Internet]. 2021 Sep 3 [cited 2024 Apr 23];29(1):6. Available from: https://doi.org/10.1007/s00159-021-00134-7

37. Brownlee J. Train-Test Split for Evaluating Machine Learning Algorithms [Internet]. Machine Learning Mastery. Available from: https://machinelearningmastery.com test-split-for-evaluating-machine-learning-algorithms/

38. Shah R. Tune hyperparameters with gridsearchcv [Internet]. Analytics Vidhya. 2021 [cited 2024 Apr 23]. Available from: https://www.analyticsvidhya.com/blog/2021/06 hyperparameters-with-gridsearchcv/

39. Taylor M. Topcat [Internet]. University of Bristol; [cited 2023 Nov 1]. Available from: https://www.star.bris.ac.uk/ mbt/topcat/

40. Kretschmar P, Fürst F, Sidoli L, Bozzo E, Alfonso-Garzón J, Bodaghee A, et al. Advances in understanding high-mass x-ray binaries with integraland future directions. New Astronomy Reviews [Internet]. 2019 Dec 1 [cited 2024 Apr 23];86:101546. Available from: https://www.sciencedirect.com/science/article/pii/S13876

41. Low-mass x-ray binary stars - an overview — sciencedirect topics [Internet]. [cited 2024 Apr 23]. Available from: https://www.sciencedirect.com/topics/physics-and-astronomy/low-mass-x-ray-binary-stars

42. Cataclysmic variables [Internet]. [cited 2024 Apr 20]. Available from: https://heasarc.gsfc.nasa.gov/docs/objects/cvs/cvstext.html

43. Lang D, Hogg DW, Mierle K, Blanton M, Roweis S. Astrometry. Net: blind astrometric calibration of arbitrary astronomical images. AJ [Internet]. 2010 Mar [cited 2024 Apr 23];139(5):1782. Available from: https://dx.doi.org/10.1088/0004-6256/139/5/1782

44. Postnov KA, Yungelson LR. The evolution of compact binary star systems. Living Rev Relativ [Internet]. 2014 May 5 [cited 2024 Apr 23];17(1):3. Available from: https://doi.org/10.12942/lrr-2014-3

45. Iben I, Livio M. Common envelopes in binary star evolution. Publications of the Astronomical Society of the Pacific [Internet]. 1993 [cited 2024 Apr 23];105(694):1373–406. Available from: https://www.jstor.org/stable/40680203

46. Soltis J, Casertano S, Riess AG. The parallax of centauri measured from gaia edr3 and a direct, geometric calibration of the tip of the red giant branch and the hubble constant. ApJL [Internet]. 2021 Feb [cited 2024 Apr 23];908(1):L5. Available from: https://dx.doi.org/10.3847/2041-8213/abdbad