Optimizing Hyperparameters in Star-Galaxy-Quasar Classification Models: A Genetic Algorithm Approach

M.Sc Data Science Dissertation

16 August 2023

Pratik Barve
Student Number: C00290846

Supervisor: Paul Barry

Master of Science in Data Science
Course Code: KCDAR_M_Y5
Department of Computing
South East Technological University, Carlow
Ireland

Word Count: 3453

Contents

Chapter 1

Literature Review

1.1 Previous works

The development of big astronomical surveys has greatly increased the importance of automatic data classification and processing. The separation of photometric catalogs into stars and galaxies must be automated because there is simply too much data for human specialists to manually classify (Kim and Brunner, 2016). These large surveys are aimed at generating astronomical catalogues which then can be used for further studies of the objects. This makes accurate classification a crucial step.

There are 3 major categories of galaxies according to their morphologies as per Hubble galaxy classification scheme (Hubble, 1926). Any manual classification and analysis of large datasets requires a considerable amount of time and effort. The categorization of galaxies is difficult and imprecise due to the complicated nature of galaxies and the characteristics of the pictures (Khalifa et al., 2018). For scenarios with several categories, deep learning has shown notable results and significantly improved visual detection and identification. The Convolutional Neural Network (CNN) is the most prevalent type of deep, feed-forward neural network and one of the most renowned deep learning algorithms (Khalifa et al., 2018). The term ”feed-forward” refers to the flow of information through the network in a single direction, i.e., moving forward from the input layer to output layer without any feedback connections.

Instead of using image data from the astronomical surveys, recently, accurate stellar object classification using photometric data has gained much popularity (Wierzbiński et al., 2021). Machine Learning (ML) algorithms consists of the model parameters and hyper parameters. Hyperparameter is a configuration that is external to the model and whose value cannot be determined from the data. With increasing complexity of model, the number of hyperparameter configurations to be determined increases (Cui and Bai, 2019). One of the current practices to set hyperparameters is manual search, that is basically relying on human intuition and experience. The hyperparameters that work on one set of data are not guaranteed to work on another set of data (Young et al., 2015). One other practice to set hyperparameters is grid search in which a range of values of hyperparameters is used to generate a grid space and every value is tried to find the best set of values. To make this process faster, instead of trying all the values in the search space, random search can be implemented. This method does not guarantee the most optimum set of values and is dependent on the computing resources available.

1.2 ML Algorithms

1.2.1 Support Vector Machines (SVMs)

The support-vector network follows the following theory: it maps the input vectors into a high-dimensional feature space using a non-linear mapping that is predetermined in advance. A linear decision surface with unique characteristics is built in this area to ensure the network's excellent generalization ability (Cortes and Vapnik, 1995). Support Vector Machines (SVMs) is a supervised machine learning algorithm used to solve classification and regression problems. The core idea is to find the hyperplane in the feature space to separate two classes (Jin et al., 2019). Like fitting a line in 2D to separate 2 classes, hyperplane separates multidimensional feature space. An optimal hyperplane, see Figure 1.1 can be defined as a linear decision function with maximum margin between the vectors of two classes (Cortes and Vapnik, 1995).

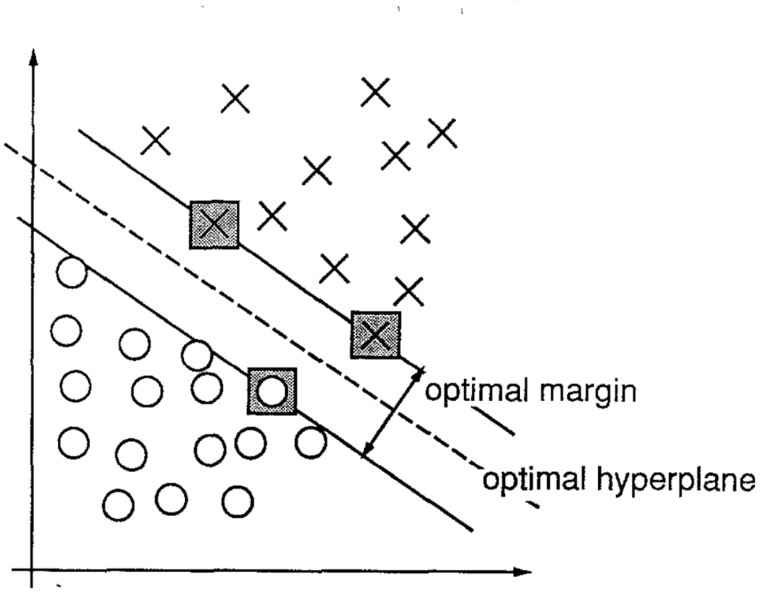

Figure 1.1: The support vectors, marked with grey squares, define the margin of largest separation between the two classes (Cortes and Vapnik, 1995).

One just has to account for a tiny portion of the training data, the so-called support vectors, which define this margin, in order to create such ideal hyperplanes (Cortes and Vapnik, 1995). SVM has been used widely for classification in astronomy (Zhang et al., 2012) (Kim and Brunner, 2016). In SVM, the hyperparameters are, Kernel, C and Gamma (Buitinck et al., 2013).

- **Kernel**: SVM can use different kernel functions to transform the input data into a higher-dimensional space. The kernel function determines the type of decision boundary created by SVM, such as linear, polynomial, or radial basis function (RBF) kernels.
- **C**: This hyperparameter controls the trade-off between maximizing the margin (distance between the decision boundary and the nearest data points) and minimizing the classification errors. A smaller value of C allows more misclassifications but increases the margin, while a larger value of C reduces the margin but minimizes misclassifications.

- **Gamma**: This hyperparameter is specific to the RBF kernel and determines the influence of each training example. A higher value of gamma leads to a more complex decision boundary that can fit the training data more precisely. However, it may also result in overfitting if the value is too large.

1.2.2 k-Nearest Neighbours (kNNs)

The k-nearest neighbors algorithm (k-NN) is a simple and intuitive machine learning method used for classification or regression tasks. It is considered non-parametric because it doesn't make any assumptions about the underlying data distribution (Cover and Hart, 1967). In simpler terms, k-NN works by looking at the "neighbors" of a data point to make predictions. When given a new, unlabeled data point, k-NN compares it to the labeled data points in its training set. It measures the distance between the new data point and the labeled data points using a distance metric, such as the straight-line distance between their feature values.

The "k" in k-NN refers to the number of neighbors that will be considered. For example, if k is set to 3, the algorithm will look at the three nearest labeled data points to the new data point. Once the algorithm has identified the k-nearest neighbors, it takes a majority vote (in the case of classification) or computes the average (in the case of regression) of their labels to assign a label or value to the new data point.

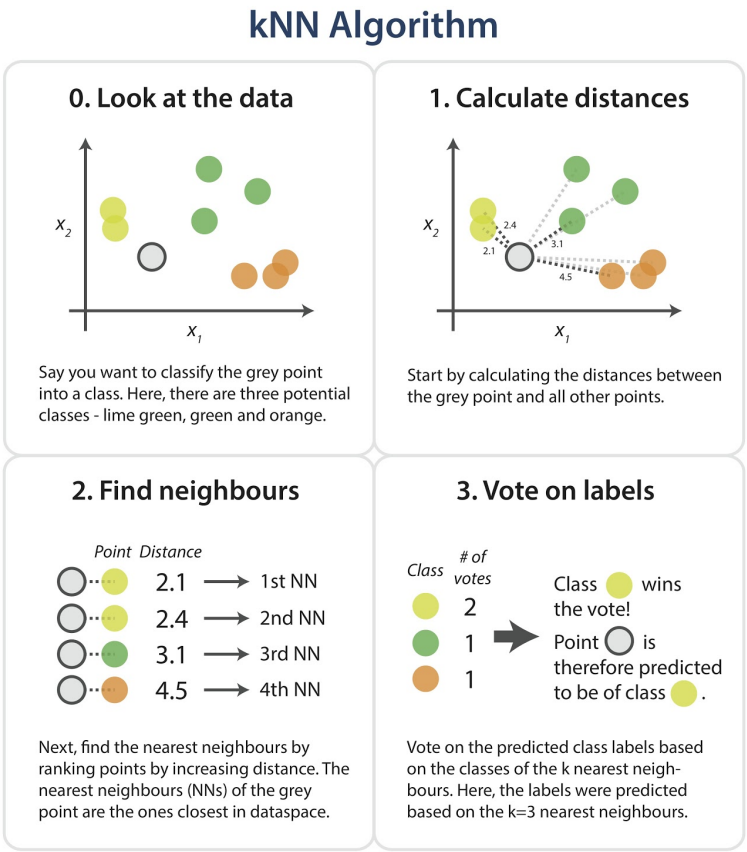

Figure 1.2: Steps of kNN algorithm (Latysheva, 2016).

Due to these charactersitics kNNs have been used for astronomical object classfication by Zhang et al. (2012). The hyperparameters in k-nearest neighbours are:

- **Number of neighbors (k)**: This hyperparameter determines the number of nearest neighbors in the training data that are considered when making a prediction for a new data point. A larger value of k considers more neighbors, which can smooth out noise but might result in a loss of local detail (Buitinck et al., 2013).

- **Distance metric**: kNN uses a distance metric to measure the similarity or dissimilarity between data points. Common distance metrics include Euclidean distance and Manhattan distance. The choice of distance metric depends on the nature of the data and the problem at hand.

1.2.3 Random Forest

Random forests, also known as random decision forests, are an ensemble learning method widely utilized in various domains, including classification and regression tasks. The technique involves constructing an ensemble of decision trees during the training phase.

In the context of classification tasks, random forests aggregate predictions from multiple decision trees to determine the final class assignment. Each decision tree within the forest independently evaluates a subset of features and a random subset of the training data. By incorporating these randomized components, the random forest fosters diversity among the individual trees, mitigating the risk of overfitting and enhancing robustness.

The underlying principle of random forests centers around the notion that aggregating the predictions of multiple decision trees can yield superior performance compared to any single tree. By combining the outputs of diverse decision trees, the ensemble model is capable of mitigating individual tree biases and errors, resulting in improved generalization and enhanced predictive power (Ho, 1995).

Due to these characteristics random forests have wide applicability in astronomical object classification (Becker et al., 2020).

The hyperparameters in Random Decision Forests are (Buitinck et al., 2013):

- **Number of trees**: Random Forest creates an ensemble of decision trees, and this hyperparameter determines the number of trees in the ensemble. A larger number of trees can improve the model's accuracy but increases the computational cost.

- **Maximum depth**: This hyperparameter restricts the depth of each decision tree in the Random Forest. Controlling the maximum depth helps to prevent overfitting. A deeper tree can capture more complex relationships in the data, but it may also lead to overfitting if the depth is not properly constrained.

- **Number of features**: At each node of a decision tree in Random Forest, a random subset of features is considered for splitting. This hyperparameter determines the number of features to be considered. It provides randomness to the model and reduces the correlation between trees.

1.3 Sloan Digital Sky Survey (SDSS)

The Sloan Digital Sky Survey (SDSS) is a large scale multi spectral imaging and spectroscopic redshift survey using a dedicated 2.5 meter wide-angle optical telescope at Apache Point Observatory in New Mexico, USA. It began operation in the year 2000 and since then have been taking continuous wide angle high resolution images of the sky. The telescope used a camera consisting of 30 charge-coupled device (CCD) chips with a resolution of 2048 $\times$ 2048 each. The chips were arranged in 5 rows with 6 in each row. Each row observes the space through various optical filters (u, g, r, i, z) with different wavelengths $u = 354$ nm, $g = 475$ nm, $r = 622$ nm, $i = 763$ nm, and $z = 905$ nm. It is estimated that every night SDSS captures about 200 GB of data. I will be using data from the latest data release i.e., the SDSS DR18 (Data Release 18) (Almeida and et. al, 2023)

The data is publicly available through the CAS and skyserver service provide by the SDSS consortium. Through the sql search service, I downloaded the top 500 thousand results with columns and filters recommended by SDSS.

Column Name	Description
objid	The unique identifier for the object in the PhotoObj table.
ra	The right ascension of the object in degrees.
dec	The declination of the object in degrees.
u, g, r, i, z	The magnitudes of the object in different bands (ultraviolet, green, red, near-infrared, and infrared).
run	The specific run in which the object was observed.
rerun	The rerun number for the run.
camcol	The camera column number.
field	The field number.
specobjid	The unique identifier for the object in the SpecObj table.
class	The classification of the object (e.g., star, galaxy, quasar).
z as redshift	The redshift value of the object, indicating its recession velocity.
plate	The plate number on which the object was observed.
mjd	The Modified Julian Date (MJD) of the observation.
fiberid	The fiber identification number.

Of these columns the correlation between class of object is highest with columns like the color bands (u,g,r,i,z) and redshift. The optical filters used in the camera setup have specific wavelengths and are represented by the labels u, g, r, i, and z. These filters play a significant role in observing the space and capturing different aspects of the electromagnetic spectrum.

Filter u (354 nm): This filter allows the observation of ultraviolet light, enabling the detection of objects or phenomena that emit or interact strongly with ultraviolet radiation.

Filter g (475 nm): This filter captures light in the blue-green part of the spectrum. It is particularly useful for studying objects that emit light in this range, such as certain types of stars and galaxies.

Filter r (622 nm): This filter is sensitive to red light, enabling the detection of objects that emit or reflect red wavelengths. It is often used to study the characteristics of red stars, galaxies, and certain astronomical phenomena.

Filter i (763 nm): This filter is designed to capture near-infrared light. It allows the observation of objects that emit or interact strongly with infrared radiation, providing insights into phenomena such as star formation, dust clouds, and certain celestial objects that emit primarily in the infrared.

Filter z (905 nm): This filter extends further into the infrared part of the spectrum. It enables the detection of objects that emit or interact strongly with longer-wavelength infrared radiation, providing valuable information about objects such as distant galaxies, quasars, and other infrared-emitting sources.

By using these specific optical filters with distinct wavelengths, astronomers can gather data from different regions of the electromagnetic spectrum, allowing for a comprehensive study of celestial objects and phenomena across a wide range of wavelengths.

These optical filters play a crucial role in star-galaxy-quasar classification. By observing celestial objects through different filters, astronomers can gather information about their spectral properties and distinguish between different types of objects.

The filters u', g', r', i', and z' capture light at specific wavelengths, allowing astronomers to study the characteristics and emission patterns of stars, galaxies, and quasars. For example, the u' filter enables the detection of blue light, which is often associated with young, hot stars. The r' and i' filters are sensitive to red and near-infrared light, respectively, which can provide insights into the properties of red stars, galaxies, and objects emitting in the infrared spectrum. The z' filter extends further into the infrared, enabling the observation of distant galaxies and quasars that emit predominantly in the longer-wavelength infrared range.

By analyzing the light captured through these filters and combining the observations with other techniques, astronomers can classify celestial objects as stars, galaxies, or quasars based on their distinct spectral signatures. This classification helps in understanding the nature and evolution of these objects, unraveling the mysteries of the universe.

1.4 Genetic Algorithms (GAs)

According to Darwin's rule of natural selection, organisms with phenotypes that are well matched to their immediate environment will eventually have a higher probability of reproducing and/or surviving than other, less suitable species (Dodson, 1976). In the following generations, only those phenotypes prevails which are more suited to the environment. Mutation of genes can cause the organism to either have better characteristics or worse characteristics. But natural selection will eventually carry forward those mutations which are better.

Genetic Algorithm (GA) is a computational model which takes inspiration from Darwin's law of natural selection. The terminologies used to describe such types of algorithms are analogous to the biological ones. A population-based search technique called the Genetic Algorithm (GA) creates new populations by repeatedly using genetic operators or randomization functions (Katoch et al., 2021).GA

effectively searches an encoded parameter space by using the objects from the population and the randomization technique (Liu, 2019).

A population is initialized using random variables (traits/characterstics), a set of variables is referred to as chromosomes. The main components of GA are chromosomal representation, crossover, fitness selection, mutation, and fitness function computing. Until offsprings generated provide the ideal solution or the maximum number of iterations is achieved, the selection, crossover, and mutation procedures on the population are repeated. GA can be summaried in the form of a pseudo-code.

The fundamental theorem of GA is the Schema Theorem (Liu, 2019). "According to Schema theorem, the original schema has to be replaced with modified schema. To maintain the diversity in population, the new schema keeps the initial population during the early stages of evolution. At the end of evolution, the appropriate schema will be produced to prevent any distortion of excellent genetic schema." (Katoch et al., 2021) The most crucial element of genetic operators is the crossover operation. It is a recombination operator that joins pieces of the chromosomes of two parents to create offspring that have a portion of both parents' genetic makeup. (Tang et al., 1996). "The offspring generation formed by the cross-interchange of the parental chromosomes is likely to discard or destroy the excellent genetic schemas possessed by the parental individual."(Liu, 2019)

1.4.1 Genetic Operators

Genetic operators are the operators that are used during the optimisation process. See Figure 1.3 the general operators are encoding schemes, selection, crossover and mutation.

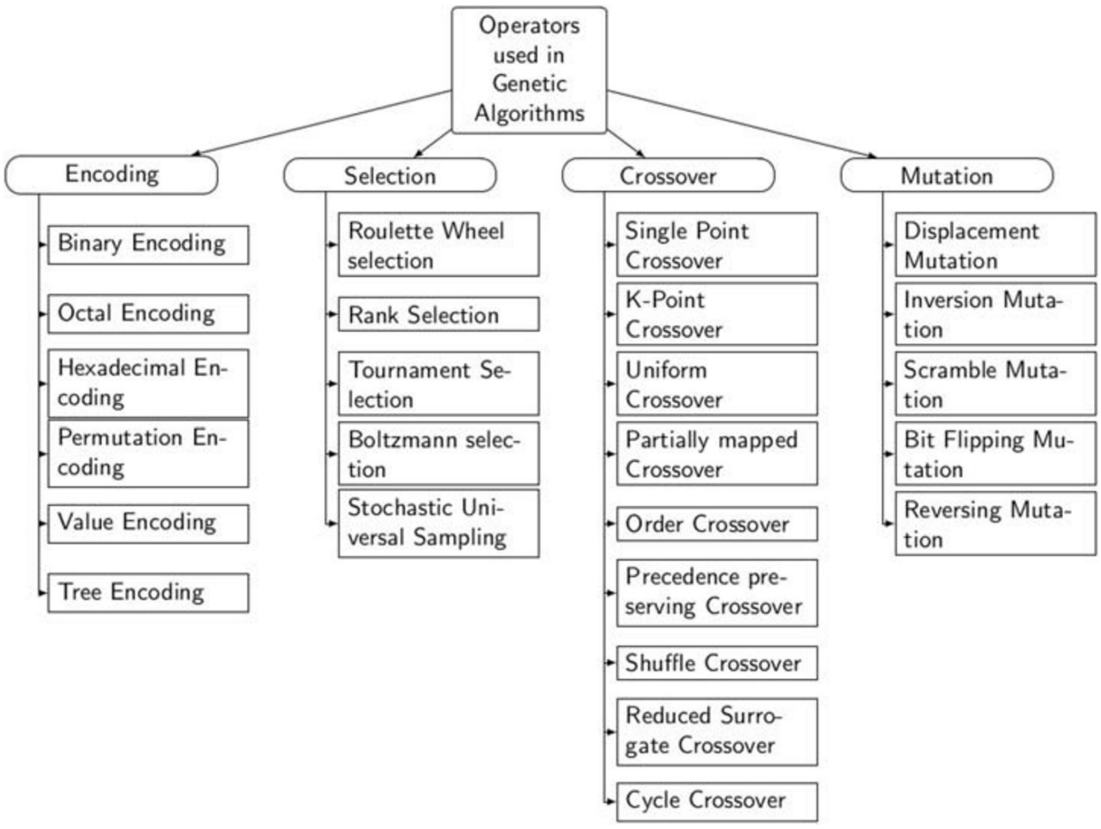

Figure 1.3: Operators used in Genetic Algorithms (Katoch et al., 2021)

Representation (Encoding Schemes)

We encode the solution set of the problem to improve the algorithm's performance. This encoded object functions analogous to a chromosome. It is not the problem in itself, but rather a collection of attempted solutions to the problem. The chromosome can be represented / encoded as binary, octadecimal, or hexadecimal values, and these can then be subjected to crossover and mutation to provide a more likely set of solutions. Aside from numerical encoding, "permutation encoding" is generally preferred in ordering problems.

Selection

In genetic algorithms, selection is a key phase that determines whether a particular string participates in reproduction. Boltzmann, rank, tournaments, roulette wheels, and universal probabilistic sampling are examples of well-known selection techniques. The roulette wheel selection method maps all possible strings to the wheel and assigns each string a space on the wheel based on the fit value. This wheel is then randomly rotated to select a specific solution involved in building the next generation (Holland, 1992) (Katoch et al., 2021).

The modified version of roulettewheel selection is ranked selection. Instead of using fitness value, it uses rankings. Brindle introduced the tournament selection approach in 1983, in which the individuals are chosen in pairs from a stochastic roulette wheel based on their fitness scores. Following selection, those with a greater fitness value are added to the pool of the following generation. (Holland, 1992)

The existing roulette wheel selection method is extended by stochastic universal sampling (SUS). It chooses an individual at regularly spaced intervals from a list of individuals from a generation, starting at a random position. All individuals have an equal chance of being chosen to take part in the following generation's crossover (Katoch et al., 2021).

Crossover

In order to create offspring, crossover operators combine the genetic information of two or more parents.

Many GA practitioners believe that the crossover operator is what really sets the GA apart from all other optimization algorithms. There are several crossover operation variants offered, with a single-point crossover being the most basic. Based on the selection methodology, the parents are randomly chosen. The regions of the two chromosomes beyond the crossover point are exchanged to create the offspring, where the crossover point is determined by a random process (Tang et al., 1996).

Similar to single-point crossover, multipoint crossover uses m crossover sites that are randomly selected without duplication. The crossover operator that is most often employed is partially matched crossover (PMX). This operator outperforms the majority of the other crossover operators in terms of performance (Katoch et al., 2021).

Mutation

An operator that modifies the chromosome is called mutation. The mutation can take place both on a global and local level. The procedure randomly changes the value of a string position occasionally (and typically with low probability p) (Tang et al., 1996). The three most well-known mutation operators are scramble mutation, simple inver-sion, and displacement.

The displacement mutation (DM) operator moves an individual solution's substring within itself. To ensure that both the final solution and a random displacement mutation are legitimate, the location is randomly selected from the substring that is being used for displacement. Exchange mutation and insertion mutation are two types of DM variations. A portion of a unique solution is either swapped with another portion or inserted in a different position in exchange mutation and insertion mutation operators, respectively. (Jebari, 2013)

The SIM (simple inversion mutation operator) inversion operator flips the randomly chosen string and positions it in a random spot. Between any two given places in a single solution, the substring is reversed using the SIM (Jebari, 2013). The scramble mutation (SM) operator arranges the components of each individual solution in a random order and determines whether the freshly created solution's fitness value has increased or decreased (Jebari, 2013).

Chapter 2

Methodology

2.1 EDA/Preprocessing

The first step is to obtain the dataset from the skyserver service provided by the SDSS consortium (Almeida and et. al, 2023). Using skyserver 500 thousand astronomical object dataset has been downloaded using recommended sql query and filters. Initial exploratory data analysis (EDA) has to be performed to gain deeper insights into the dataset and obtain correlation between the object attributes. The correlation matrix shows high correlation between the u,g,r,i and z filters of the telescope.

Figure 2.1: Correlation matrix between the color bands of the telescope.

The preprocessing of data will require some cleaning, handling missing values if any, feature normalisation or scaling using standard functions like StandardNormalScaler from sklearn package. Principal Component Analysis (PCA) can also be performed to reduce the feature space and increase training speeds. The next step is to split the dataset into training and testing sets, ensuring that the class distribution is maintained in both sets.

2.2 Establish baseline performance

Before implementing GAs to the models it is important to establish the baseline performance of the models using the default hyperparameters. For this thesis, I have selected 5 machine learning models which are widely used in machine learning classification tasks - SVMs, kNNs, Random Forests, Decision Trees, Gradient Boosting Algorithms.

Using the default parameters the models will be trained and the performance will be evaulated using the standard evaluation metrics such as accuracy, precision, recall and F1-score. These metrics will be saved for comparison with models trained after optimising hyperparameters.

2.3 Implementing Genetic Algorithms

To implement genetic algorithms in tuning hyperparameters, DEAP - Distributed Evolutionary Algorithms in Python, library can be used. The functions from this library seamlessly integrate with machine learning models in sklearn library.

1. Define the Hyperparameter Space: Determine the range or set of possible values for each hyperparameter that needs to be tuned in scikit-learn algorithms. It is important to set a hyperparameter space because of limited computing resources.

2. Define the Fitness Function: Create a fitness function that evaluates the performance of a model with a specific set of hyperparameters. The fitness function should interface with scikit-learn by training and evaluating the model on the desired dataset.

3. Set Up the Genetic Algorithm: Use DEAP to set up the genetic algorithm by defining the population, genetic operators (crossover and mutation), selection mechanisms, and termination criteria.

4. Integrate with scikit-learn: Within the genetic algorithm framework, sklearn will be used to train and evaluate the models with different hyperparameter settings based on the genetic algorithm's suggestions.

5. Retrieve the Best Hyperparameters: Once the genetic algorithm completes, the best hyperparameters can be extracted from the fittest individual(s) and then can be used to train the final model on the full dataset.

2.4 Alternate optimisation methods

There are other methods of tuning hyperparameters like grid search and random search. It will be useful to train models using these methods and obtain metrics for better comparison of which method works best for the application of star-galaxy-quasar classification.

ELABORATE MORE ON RANDOM SEARCH AND GRID SEARCH

2.5 Comparison

Using the metrics from models trained using default hyperparamters, genetically optimised hyperparameters and alternate optimisation techniques, draw conclusions as to which technique works best for our purpose.

Bibliography

Almeida, A. and et. al (2023), 'The Eighteenth Data Release of the Sloan Digital Sky Surveys: Targeting and First Spectra from SDSS-V', *arXiv e-prints* p. arXiv:2301.07688.

Becker, I., Pichara, K., Catelan, M., Protopapas, P., Aguirre, C. and Nikzat, F. (2020), 'Scalable end-to-end recurrent neural network for variable star classification', *Monthly Notices of the Royal Astronomical Society* **493**(2), 2981–2995.

Buitinck, L., Louppe, G., Blondel, M., Pedregosa, F., Mueller, A., Grisel, O., Niculae, V., Prettenhofer, P., Gramfort, A., Grobler, J., Layton, R., VanderPlas, J., Joly, A., Holt, B. and Varoquaux, G. (2013), API design for machine learning software: experiences from the scikit-learn project, *in* 'ECML PKDD Workshop: Languages for Data Mining and Machine Learning', pp. 108–122.

Cortes, C. and Vapnik, V. (1995), 'Support-vector networks', *Machine Learning* **20**, 273–297.

Cover, T. and Hart, P. (1967), 'Nearest neighbor pattern classification', *IEEE Transactions on Information Theory* **13**(1), 21–27.

Cui, H. and Bai, J. (2019), 'A new hyperparameters optimization method for convolutional neural networks', *Pattern Recognition Letters* **125**, 828–834.

Dodson, M. (1976), 'Darwin's law of natural selection and thom's theory of catastrophes', *Mathematical Biosciences* **28**(3), 243–274.

Ho, T. K. (1995), Random decision forests, *in* 'Proceedings of 3rd International Conference on Document Analysis and Recognition', Vol. 1, pp. 278–282 vol.1.

Holland, J. H. (1992), 'Genetic algorithms', *Scientific American* **267**(1), 66–73.

Hubble, E. P. (1926), 'Extragalactic nebulae.', *ApJ* **64**, 321–369.

Jebari, K. (2013), 'Selection methods for genetic algorithms', *International Journal of Emerging Sciences* **3**, 333–344.

Jin, X., Zhang, Y., Zhang, J., Zhao, Y., Wu, X.-b. and Fan, D. (2019), 'Efficient selection of quasar candidates based on optical and infrared photometric data using machine learning', *Monthly Notices of the Royal Astronomical Society* **485**(4), 4539–4549.

Katoch, S., Chauhan, S. S. and Kumar, V. (2021), 'A review on genetic algorithm: past, present, and future', *Multimedia Tools and Applications* **80**, 8091–8126.

Khalifa, N. E., Hamed Taha, M., Hassanien, A. E. and Selim, I. (2018), 'Deep galaxy v2: Robust deep convolutional neural networks for galaxy morphology classifications', pp. 1–6.

Kim, E. J. and Brunner, R. J. (2016), 'Star-galaxy classification using deep convolutional neural networks'.

Latysheva, N. (2016), 'Implementing your own k-nearest neighbor algorithm using python', https://www.kdnuggets.com/2016/01/implementing-your-own-knn-using-python.html. Accessed: 15 June 2023.

Liu, D. (2019), 'Mathematical modeling analysis of genetic algorithms under schema theorem', *Journal of Computational Methods in Sciences and Engineering* **9**, 131–137.

Tang, K., Man, K., Kwong, S. and He, Q. (1996), 'Genetic algorithms and their applications', *IEEE Signal Processing Magazine* **13**, 22–37.

Wierzbiński, M., Pławiak, P., Hammad, M. and Acharya, U. R. (2021), 'Development of accurate classification of heavenly bodies using novel machine learning techniques', *Soft Computing* **25**, 7213–7228.

Young, S. R., Rose, D. C., Karnowski, T. P., Lim, S.-H. and Patton, R. M. (2015), 'Optimizing deep learning hyper-parameters through an evolutionary algorithm', *Proceedings of the Workshop on Machine Learning in High-Performance Computing Environments* .

Zhang, Y., Zhao, Y., Zheng, H. and bing Wu, X. (2012), 'Classification of quasars and stars by supervised and unsupervised methods', *Proceedings of the International Astronomical Union* **8**, 333–334.