Multi-Class Classification using Logistic Regression: Stellar Classification

Identifying stars, quasars, and galaxies is an important task in astronomy. It is a common task performed by scientists and requires a lot of effort. The availability of more compute resources and the advancement in machine learning and AI has made it possible to reduce the complexity of this task. Trained machine learning models are able to perform these type of classification tasks in less than a minute and give higher precision scores. This assignment aims to classify galaxies, quasars, and stars based on attributes with continuous values. The logistic regression model is used to build the classifier and two different methods for performing multi-class classification are evaluated in this assignment. The One-vs-One and One-vs-Rest methods are used to allow logistic regression model to perform multi-class classification. Several pre-processing techniques are also implemented to get the optimal results and achieve the best accuracy of 94% using the One-vs-One method.

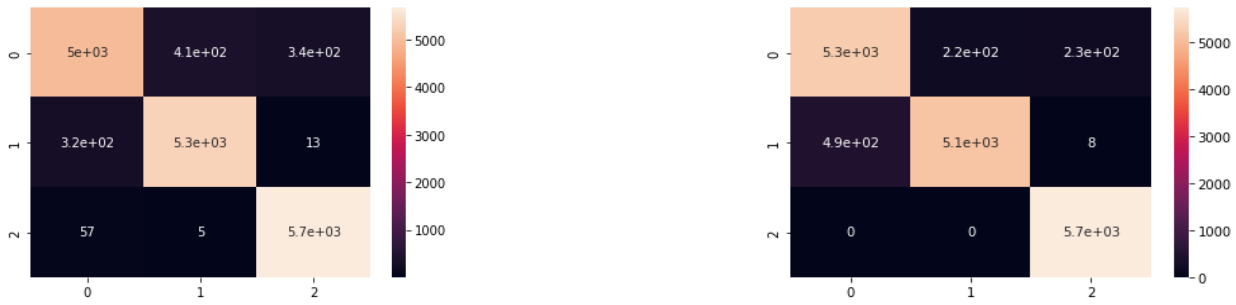
The dataset used for this assignment is publicly available on Kaggle and it is called the "Stellar Classification Dataset- SDSS17" [1]. It is a large dataset with 18 attributes and 3 target values and 100,000 instances to examine. The different attributes include: Object ID, Alpha (Right Ascension Angle at J2000 epoch), Delta (Declination Angle at J2000 epoch), u (ultraviolet filter), g (green filter), r (red filter), I (near infrared filter), z (infrared filter), run_id, rerun_id, cam_col (camera column), field_ID, spec_obj_ID, redshift value, plate, MJD (modified julian date), and fiber_ID. The dataset is complex and highly imbalanced with 59445 samples belonging to the galaxy class or class 0, 21594 samples belonging to star class or class 1, and 18961 samples belonging to quasar class or class 2. To balance the dataset under sampling was performed and all the samples were reduced to 18961 for each target variable. The dataset was visualized using a histogram from the seaborn library in order to analyze the class distribution. Additional data preprocessing and feature reduction was performed as the first few runs of the model resulted in a very poor accuracy, precision, recall, and f1 score of 59%. All the features containing ID were dropped and only the relevant features were kept. After performing feature reduction, there were only 11 features left for training the model and these instances were also scaled using the standard scaler

function from the sklearn library. Next, the dataset was split into training and test set. The training set contains 70% of the samples and the test set contains 30% of the 100,000 instances.

To handle the imbalanced dataset, the stratify method was also used when the dataset was split into training and test set, but this technique did not seem to improve the results during the first few attempts of improving the model. As an alternative, the under-sampling method was used. It is also possible to investigate the effects of over-sampling in the future, however there is always a risk of overfitting the data specially when it is as skewed as the dataset used in this assignment.

The logistic regression model from the sklearn library as well as the OneVsOneClassifier and OneVsRestClassifier were imported and used for building and training the model on the training data. The test set was used for predicting the dataset and the model was evaluated using the actual labels versus the predicted labels. To evaluate the model, the accuracy, recall, precision, and f1 scores were measured. The confusion matrix was also analyzed. The first model which is using the one versus rest method produced an accuracy, precision, recall, and f1score of 93%. The second model using the one versus one method produced a 94% accuracy, precision, recall, and f1score. The confusion matrix for both models is illustrated below. Based on these results it can be concluded that the one versus one method produced the optimal results, however it did require a lot of pre-processing in order to achieve these results. For such complex problems, it is better to use machine learning models that are capable of performing multi-class classification or use neural networks as they are able to perform feature reduction on their own.

Image I: Confusion Matrix for One-vs-Rest on the Left and One-vs-One on the Right



References:

[1] Fedesoriano, "Stellar classification dataset - SDSS17," *Kaggle*, 15-Jan-2022. Available: https://www.kaggle.com/datasets/fedesoriano/stellar-classification-dataset-sdss17.