|  |  |
| --- | --- |
| **DATA 430 Technical Report Assignment 2: Bayesian Classification** | **Sulchan Yoon** |
| **Classification of Stars, Galaxies, and Quasars** | |
| **URL to dataset: https://www.kaggle.com/datasets/fedesoriano/stellar-classification-dataset-sdss17** | |

This template should be used in conjunction with the assignment instructions. The size of the text area below will expand to the length of your response; the area should not be interpreted as a required or suggested length of response. Responses within the text area should be single spaced with Times New Roman 12pt font. The body of the document will likely be 6-9 pages, not including the Appendix; length may vary depending on specifics of the analysis and the dataset. As needed, APA format in-text citations should be included, along with a full references list at the end of the document.

|  |
| --- |
| **Overview** |
| **Problem Domain**: give some background and context about the problem domain (application area). For instance, if you are doing the analysis for predicting heart disease, provide some context about the disease and include some interesting statistics about it. Also, discuss how the method is relevant for the chosen problem. |
| Astronomy and Space has been a major topic in recent years with SpaceX bringing exploration much more into the public domain. Observations through a variety of tools and instruments has allowed individuals to classify stars based on their spectral characteristics. With the data gathered through these tools, data scientists, and astronomers have been able to group galaxies into a variety of classifications, allowing us to find those that may be similar to the Milky Way that we are in. Galaxies and star clusters such as the popular Andromeda Galaxy and Pleiades Star Cluster have been those that came about these classification methods.  This is by no means the only method of classification, however it is one set of tools scientists and researchers are able to continue understanding the vast universe in search of potentially a new livable planet. In an Astrophysics sense, we can use these metrics to support how actions and reactions occur as it relates to our known understanding of the Laws of Physics. In Astrometry terms, determining classifications through different filters and wavelengths help map out celestial bodies to better understand things such as a black hole, or dark matter. We also learn that our own galaxy, the Milky Way in its current form is approximately 100,000 light years across, and the closest solar system similar to ours would be Andromeda Galaxy at around 2 Million light years away.  With this information, we want to find a more streamlined method to classify objects and would look to have a predictive machine learning algorithm to support the need. |
| **Objective**: clearly state the objective of the analysis in relation to the kind of algorithm you are employing. Use specific language as to what question(s) you are trying to answer using the specific analysis/modeling type. |
| The purpose of this report is to analyze the given information to find trends of the information as part of exploratory data analysis. We will then go deeper into the data and attempt to produce a viable machine learning algorithm to predict class types. Our first attempt will be through Logistic Regression, then we compare that to other methods such as decision trees, and Bayesian Classifications. As part of our secondary attempt, we are looking to focus in on the Naive Bayes method.  Overall, our purpose is to use these data elements in a way to find high value machine learning techniques to support astronomical research. Our predictive analysis is focused on trying to appropriately classify a celestial body (dependent variable) using the different wavelengths readings, body metrics (i.e. ascension), plate ID information (independent variables). Our first attempt of predictive analysis will be handled via logistic regression. As part of project 2, we will focus on Naive Bayes algorithms.  Naive Bayes is one of the simpler machine learning algorithms with high effectiveness. Typically, Naive Bayes is used for solving multi-class prediction problems, especially when there is independence of features. Though our dataset has a significant amount of data points, a Naive Bayes algorithm requires much less training data to be effective and perform well. |
| **Analysis** |
| **Exploratory Analysis**: describe the data including the source, the collection method, and variables. Perform exploratory analysis. Also, select few key variables (including the target variable for supervised learning) and study their distributions using plots such as histograms, box plot, bar chart, etc. |
| This data has been provided by the Sloan Digital Sky Survey’s 4th phase (SDSS-IV). The information presented will be based on the Data Release 17 published in 2022 as the final data release. The data has also been peer reviewed and reduced to 100,000 observations for Kaggle use. The main extractions come from the Mapping Nearby Galaxies at Apache Point Observatory survey, bringing in over 10,000 nearby galaxy observations. Data from the DR17 also includes all information from the prior releases and includes information from Apache Point Observatory Galactic Evolution Experiment 2 survey pulling in 650,000 star readings.  Data points captured through this survey have been captured through ground stations. There are a variety of different instruments that have been capturing data points such as the Hubble Space Telescope which gives us a different understanding of the universe. The value of capturing information on ground stations is to see these planets are they relate to Earth, whereas instruments in space can capture with a point of view not equal to Earth.  The total dataset includes 18 total variables, however as we go through and review each of the dataset values, we recognize that some variables are not as valuable due to its variance within each classification. Our initial exploratory data analysis includes descriptive statistics. A lot can be seen with the ID variations and noted that overall IDs are not valuable data points. We remove all ID variables and review the remaining variables in more detail and through visualizations.  We go through and create kernel density estimate (kde) graphs of each variable to find those that may have differences in its distribution. An example graph of the kde plot is shown below. You can find all graphs in the appendix along with the functions used to create the graphs.    Through each of the graphs, we find those that result in variances that we may consider to be unique to each type of Class. This will be used to help create the proper algorithm using the Logarithmic Regression in Python. Our overall result of variables we will use are shown below in the chart of Points of Interest.  Points of Interest   |  |  | | --- | --- | | Variable | Definition | | 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 | | Z | Infrared Filter | | Class | Object Class | | Redshift | Redshift Value on the increase in wavelength | | Plate | Plate ID in SDSS | | MJD | Modified Julian Date | |
| **Preprocessing**: armed with the exploratory analysis, perform the necessary preprocessing, both general and specific types appropriate for the modeling type being employed. |
| During the preprocessing stage, our main focus is to get items ready for fitting into the model. The first item we take a look at is to label encode our predictions. This will allow the system to read them as 0, 1, and 2. Next we are able to confirm our X and y tables, where X will be all variables we set as independent variables and y to be the dependent. This leaves us with a single dependent (classes) which have been relabeled as 0 for Galaxy, 1 for QSO, and 2 for Star. |
| **Model Fitting**: explain the key steps and activities you perform to fit the model. Experiment (as appropriate) with parameters tuning. This is key, what separates highly accurate model from a less accurate ones is the amount of performance tuning performed. |
| Now we are in the model fitting phase of the analysis. This is where we split our data into the appropriate training and test datasets to be put into our logistical regression model. For our case, we use two different functions to help with splitting and fitting. The first is the most common (see Functions in Appendix), using the train\_test\_split function within the sklearn.model\_selection package. I ran different variations of test size ranging from standard 20% up to 40% to see the effect it had on the output. I also chose to use a random\_state of 10 such that this analysis can be reproduced at any given time. Once the training and test sets are done, we can fit the logistic regression with a multi class type equal to multinomial. Due to an issue with 100% outputs (discussed in the evaluation section), we do explore more into a Standard Scaler function from sklearn.preprocessing to attempt to correct for our potential errors. The purpose of the Standard Scaler function is to standardize features to a variance by clearing mean within the sample. That is specific to our scenario with multiple dependent variables for this Naive Bayes model. |
| **Results** |
| **Model Properties:** explain the components of the fitted model and their characteristics. Leverage functions to summarize the model properties. Also, leverage visualization as required. |
| We will focus on the standard train test split model here. Our final fitted model will use a test to train of 20/80. This means our test size is 20% of the total data. Our random state is set to 10, which is used for purposes of reproducibility. The actual value of the random state does not matter as that is what is used to determine the random split. For our fit function, we will use the x and y training set produced from the train test split function. |
| **Output Interpretation**: explain the result and interpret the final model output using terms that reflect the application area and in relation to the stated objective. This is where you check whether or not the stated objective is met. |
| For our output we use a variety of different methods to score our accuracy. The first method will be through the score function where the x and y test dataset will be inputted into the fitted model. Getting the score, we will be able to find the mean value which will give us our final accuracy of this model. Through the Naive Bayes model, we got a 100% accuracy rating all throughout with Precision and Recall also reflecting 100%. See below for our confusion matrix report.   |  |  |  |  |  | | --- | --- | --- | --- | --- | |  | Precision | Recall | F1-score | Support | | 0 | 1.00 | 1.00 | 1.00 | 11954 | | 1 | 1.00 | 1.00 | 1.00 | 3743 | | 2 | 1.00 | 1.00 | 1.00 | 4303 | | Accuracy |  |  | 1.00 | 20000 | | Macro Avg | 1.00 | 1.00 | 1.00 | 20000 | | Weighted Avg | 1.00 | 1.00 | 1.00 | 20000 |   Generally, the definition of Precision is to define those that are positive predictions and compare to what percentage is truly positive.    Recall is defined by those that are total positive, what is the percentage of predicted positive.    Due to our current model with Naive Bayes on our dataset, we are seeing both a 100% precision and 100% recall. To discuss the F-1 score, we have the mix of both recall and precision’s harmonic mean.    F-1 Score typically provides a more accurate reading on an imbalanced dataset. Again, due to issues with the sampling on our dataset, we see a 100% F-1 score. |
| **Evaluation**: employ appropriate metrics to quantitatively evaluate the performance of the fitted model. For supervised classification, this includes simple accuracy, precision & recall (or sensitivity & specificity), all of which can be generated from a confusion matrix, or ROC. |
| As part of the evaluation, we can clearly see there is a general issue with our model. Having perfect ratings in precision, recall, accuracy and such is not a real life scenario when using a split dataset. A common issue with this is that the datapoints from the split are not properly being split into a train and test set. In other words, the test is a subset of train and thus always comes out with an exact match causing an overfitting scenario.  Due to this “perfect” model situation, I did attempt to run this with model with and without a standard scaler transform. See below for the scaler set up.    However, this did not change anything with the output, with everything still showing as 100%. I also changed random state to go from 10, 15, 25, and also tried test sizes of 0.01, 0.5, and 0.99. Each of these changes did not have impact on the overall output of the model. Using this same train and test set split, I reran the previous models for Logistic Regression, and a Random Forest Classifier, and those did come out with an output similar to how they were. Logistic Regression provided a score 0.60-0.61 and Random Forest came around 0.96-0.98.  Upon review, the only other aspect I can imagine causing issues may be the imbalance of classifications (class Galaxy significantly high) may cause this issue. The Standard Scaler is to affect each of the data variables, but does not affect the output (class). Since my prediction is that the imbalance of the classifications is the cause of the issues, the Standard Scaler did not show any changes to the output of the model. |
| **Conclusion** |
| **Summary**: highlight the main findings in relation to the stated objective. You don’t need to discuss the details of the analysis and the model such as accuracy here, just focus on the key findings. |
| Overall, with our given dataset, we found that a Logistic Regression model provides an approximate accuracy score of 0.6 with a 0.39 precision, 0.37 recall, and f-1 score of 0.31. Our initial predictions would have been to see our Naive Bayes model to provide a more accurate with better stats (precision, recall, f-1) over our Logistic Regression model. In a technical stance, we did perform better with a 100% across the board, but that comes with a lot of doubt on the validity of the model. It is important to understand that having good numbers all around is not always a good thing and that as a data scientist, we should always be skeptical of our data and analysis. |
| **Limitations & Improvement areas**: discuss the limitations of the analysis and identify potential improvement areas for future work. This could be related to the data, algorithm, or a combination of the two. |
| Similar to the issues with the Logistic Regression model, there seems to be potentially a problem with splitting the data into training and test subsets. Upon reading further on significantly imbalanced datasets, there may need to be additional resampling that occurs through functions/packages such as imblearn.over\_sampling to resample. Using this will help to smooth out the data sample to provide a more reliable reading. However, there are downsides to over sampling, such as the fact that you are adding in cases that don’t necessarily exist to make up for the imbalance. |

|  |
| --- |
| **Appendix** |
| Functions:          Packages:    Plots: |

**References**

Abdurro’uf et al., The Seventeenth data release of the Sloan Digital Sky Surveys: Complete Release of MaNGA, MaStar and APOGEE-2 DATA (Abdurro’uf et al. 2022 ApJS 259, 35) [arXiv:2112.02026]

fedesoriano. (January 2022). Stellar Classification Dataset - SDSS17. Retrieved June 2023 from https://www.kaggle.com/fedesoriano/stellar-classification-dataset-sdss17.

Yoon. (2023). *Data\_430\_GC Project1* [ipynb]. Retrieved from https://github.com/sulchan/Data\_430\_GC