Stars and Quasars Classification

Pramod M N
Computer Science
PES University
Bengaluru, India
PES1201701557

Sai Prashanth R S
Computer Science
PES University
Bengaluru, India
PES1201700206

Anirudh Avadhani
Computer Science
PES University
Bengaluru, India
PES1201701526

I. GitHub Repository Link

https://github.com/saiprashanth1776/ml-2019-206-1526-1557.git

***Abstract*—Using Machine Learning for the task of Classifying stars and quasars from data obtained from Galex and SDSS photometric data. We used a decision tree based approach to achieve the same. Correctness of the classifier is measured using accuracy and other performance metrics such as precision, recall etc. Acceptable range is 91-100**

***Index Terms*—Decision Trees, Gini Index, Galex, SDSS**

II. Introduction

Quasars is derived from 'quasi-stellar radio source', they are distant objects powered by black holes billions of times bigger as compared to our sun. Stars are celestial objects which are self illuminating in nature. They have energy outputs which are much less in comparison to that of quasars. The energy creation here happens through nuclear fusion reactions. Stars in our galaxy are very close compared to the distance of quasars.

- **Supervised learning**: This approach of classification deals with labelled data, happens with the knowledge of **Ground Truth** i.e. we possess prior knowledge of what should be the target values for our samples.
- **Unsupervised learning**: This approach of classification works with the goal being to learn the inherent structure of our data without using explicitly provided labels.

Decision Tree is a flow chart like structure containing nodes(internal, leaf), branches. Internal node is used for representing a test performed on an attribute, leaf node contains a class label and outcome of a test is depicted using a branch. It Can be of both binary or Non Binary in nature.

ID3, **CART**, **C4.5** are some algorithms used for construction of a decision Tree.

Decision Tree Usage in classification: When a tuple 'X' with an unknown class is given, its attribute values are tested against the decision tree constructed and a path is derived from the root to leaf holding the class prediction for that tuple.

III. Data and Problem Statement

The Galaxy Evolution Explorer (GALEX) is an ultraviolet space telescope which operated from the year 2003 to 2012. Observations were made for sources in the FUV and NUV wavebands.

The Sloan Digital Sky Survey (SDSS) is an optical survey which observed large portions of the sky in the u,g,r,i,z wave bands and obtained the spectra of the sources so that their red-shifts could be determined as well. Data from two regions were extracted and matched.

- North Galactic Region: Data consists of 912 stars and 2027 quasars.
- Equatorial Region: Data consists of 9182 stars and 20716 quasars.

Primary Class label in our data set is the spectroscopic class label. Stars, quasars have optical images which are very similar in nature but their spectral energy distribution is not. So using this and Machine Learning models we perform the classification.

IV. Methodology

A decision tree is a classifier which is built recursively starting from the root extending to the leaf of the tree. The internal node specifically contains a question for which the answer is binary (in the present case) and helps in directing the classifier towards a direction when classifying an example. Thus, the internal nodes are called decision nodes. The leaf nodes of every tree is an answer to which class the example belongs. To solve the problem in hand, ID3 tree is used with the classification objective in mind but with the usage of gini impurity instead of information gain.

A. Working of a Decision Tree

The generation of a decision tree is by a top-down approach, which means the starting point of the tree is the root and after a few iterations, the leaf nodes are generated. The working can be stated briefly as:

- With the data in hand given a node, find the index in the data which will give the best split of the data. The index of the data is a combination of the row number and the column number in the data matrix. The split which gives the lowest gini impurity is the best gain.
- Once the index of the best split is found, perform the above mentioned split.
- Within the split, if all the elements belong to the same class, the split for that branch ends there. If it's not the case, the split data is recursively sent into the function for further splitting.

This process is carried out until all the branches terminate thus, or the maximum depth of the tree specified has been reached.

B. Gini Index

Gini index is a measure of the goodness of split of data which was performed in the second step of the process mentioned in the previous sub-section. Since the data is continuous in nature, any split gives two disjoint datasets: one where the split index has a value less than or equal to the split value and the other contrasting it.

$$gini = 1 - \Sigma(p_i)^2 \tag{1}$$

In the equation (1), n is the number of classes (equal to 2, as it only performs binary splits) gives the gini value of an index on the basis of number of zeroes and ones in the data set. To calculate the gini index of a split,

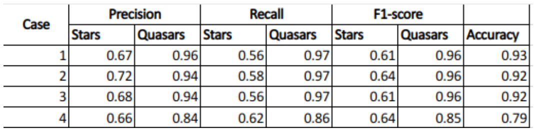
$$giniIndex = PDB * gini(PDBData) + PDA * gini(PDAData) \tag{2}$$

where **PDB** is the probability of the data being in the lower half of the data in the split while **PDA** is the probability of the data being in the upper half of the data in the split. Similarly, **PDBData** is the data of the lower half of the split and **PDAData** is the data in the upper half of the split.

V. Results and verification

Correctness of the classification model is determined by the usage of metrics such as accuracy, precision, recall. After the classification is complete the verification is performed after considering **red shift**. Maintained the train-test split ratio as 70-30

- Case 1: Catalog 1
- Case 2: Catalog 2
- Case 3: Catalog 3
- Case 4: Catalog 4

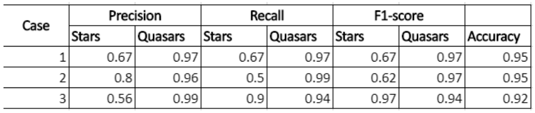


Case	Precision		Recall		F1-score		
	Stars	Quasars	Stars	Quasars	Stars	Quasars	Accuracy
1	0.67	0.96	0.56	0.97	0.61	0.96	0.93
2	0.72	0.94	0.58	0.97	0.64	0.96	0.92
3	0.68	0.94	0.56	0.97	0.61	0.96	0.92
4	0.66	0.84	0.62	0.86	0.64	0.85	0.79

Fig. 1. Catalog wise Performance Metrics

Performance metrics reported for various train-test split ratios.

- Case 1: 60-40 train-test split
- Case 2: 70-30 train-test split
- Case 3: 80-20 train-test split



Case	Precision		Recall		F1-score		
	Stars	Quasars	Stars	Quasars	Stars	Quasars	Accuracy
1	0.67	0.97	0.67	0.97	0.67	0.97	0.95
2	0.8	0.96	0.5	0.99	0.62	0.97	0.95
3	0.56	0.99	0.9	0.94	0.97	0.94	0.92

Fig. 2. Catalog 1 Performance Metrics

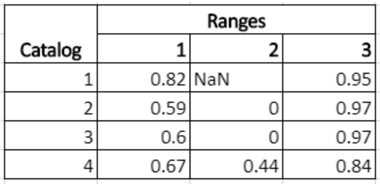
Spectrum obtained from quasars show high values of red shift indicating that are moving away at a very fast pace. In the case of stars some show blue shift (moving towards us) and others show red shift.

Range 1 is the case where z is less than or equal to 0.0033: We expect the types of samples in this range to be predominantly stars.

Range 2 is the case where z is greater than or equal to 0.004: We expect the types of samples in this range to be predominantly quasars.

Range 3 is the case where z lies in between 0.0033 and 0.004: This range of red shifts represents a grey area.

The number of samples belonging to ’Range 2’ of the data set was considerably few in number and the accuracy of the same is low owing to the above reason.



Catalog	Ranges		
	1	2	3
1	0.82	NaN	0.95
2	0.59	0	0.97
3	0.6	0	0.97
4	0.67	0.44	0.84

Fig. 3. Accuracy Measure for all Catalogs with ’red shift’ consideration

Acknowledgments

Dr. Snehanshu Saha, our guide and mentor for this project.

VI. References

1 Simran Makhija, Snehanshu Saha, Suryoday Basak, Mousumi Das. Separating Stars from Quasars: Machine Learning Investigation Using Photometric Data.

2 Bhaskar N. Patel, Satish G. Prajapati and Dr. Kamaljit I. Lakhtaria. Efficient Classification of Data Using Decision Tree.

3 Jiawei Han And Micheline Kamber,Data Mining Concept and Techniques, Copyright 2006, Second Edition

4 Quinlan J. R. (1986). Induction of decision trees. Machine Learning.

VII. GitHub Repository Link

https://github.com/saiprashanth1776/ml-2019-206-1526-1557.git