Quasar-Star Classification using a Decision Tree Regressor

Aniket Kaulavkar(PES1201700095), Rishi Ravikumar(PES1201700754), and Vishnu S Murali(PES1201701624)

Machine Learning Course(UE17CS303), PES University, Bangalore.

ABSTRACT

This report presents a Decision Tree Regressor to classify Quasars and Stars. We use decision trees to create a regression model and analyse the results predicted by the model.

Problem Statement

Binary Classification of Stars and Quasars using Machine Learning, from scratch.

Dataset Used:

The dataset used has 4 catalogues which contain 38 columns in catalogue 1 and 3 , catalogue 2 has 39 columns and catalogue 4 has 32 columns. The features used are u, g ,r ,i ,z ,extinction-u ,nuv-mag ,nuv-u,nuv-g, nuv-r, nuv-I, nuv-z, u-g, u-r ,u-I, u-z, g-r, g-I, g-z, r-I, r-z, i-z The features used are by analysing the correlation matrix shown below.

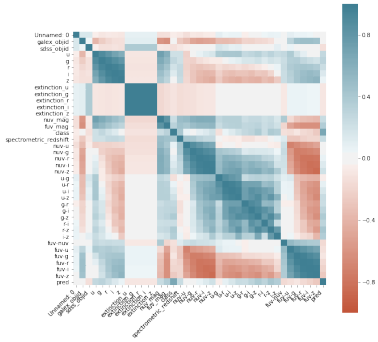

Figure 1. Correlation matrix between the features for Catalogue 1.

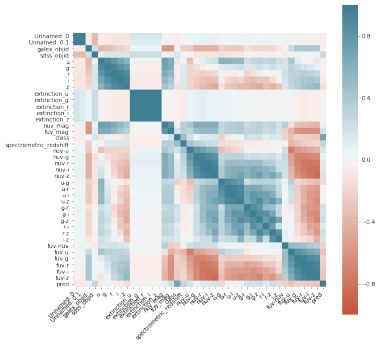

Figure 2. Correlation matrix between the features for Catalogue 2.

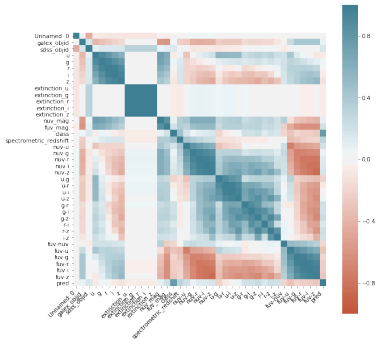

Figure 3. Correlation matrix between the features for Catalogue 3.

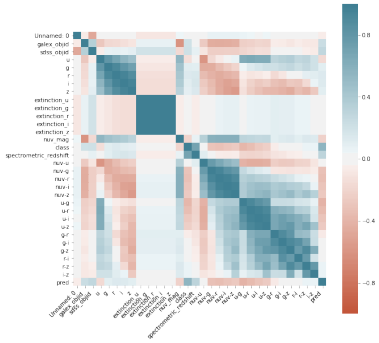

Figure 4. Correlation matrix between the features for Catalogue 4.

From the above correlation matrix we grasp that all features which have intercorrelation of **almost one** were removed from the training and testing dataset. We plotted the correlation matrix for all the four catalogues and removed the features accordingly.

ML techniques employed

Decision tree builds regression or classification models in the form of a tree structure. It breaks down a dataset into smaller and smaller subsets while at the same time an associated decision tree is incrementally developed. The final result is a tree with decision nodes and leaf nodes. A decision node has two or more branches. Leaf node represents a classification or decision (used for regression). The topmost decision node in a tree which corresponds to the best predictor (most important feature) is called a root node. Decision trees can handle both categorical and numerical data. They are used for classification and regression problems.

Basic Working: Our Node class represents one decision point in our model. Each division within the model has 2 possible outcomes for finding a solution — go to the left or go to the right. That decision point also divides our data into two sets. We create our split such that it has as low standard deviation as possible. We find the split that minimizes the weighted averages of the standard deviations which is equivalent to minimizing Root Mean Squared Error. Hence the **Cost function** used is **Root-Mean Square Error**.

$$RMSE = \sqrt{\frac{1}{m}\sum_{i=1}^{m}(y^{(i)} - h(x^{(i)}))^2}$$

The property idxs stores indexes of the subset of the data that this Node is working with. The decision (prediction) is based on the value that Node holds. To make that prediction we're just going to take the average of the data of the dependent variable for this Node. The method find-varsplit finds where should we split the data. We try to find a better feature to split on. If no such feature is found (we're at a leaf node) we do nothing. Then we use the split value found by find-better-split, create the data for the left and right nodes and create each one using a subset of the data. We create our split such that it has as low standard deviation as possible. We find the split that minimizes the weighted averages of the standard deviations which is equivalent to

minimizing RMSE. If we find a better split we store the following information: index of the variable, split score and value of the split. The score is a metric that tells us how effective the split was (note that leaf nodes do not have scores, so it will be infinity). The method find-score calculates a weighted average of the data. If the score is lower than the previous we have a better split. Note that the score is initially set to infinity -> only leaf nodes and really shallow trees have a score of infinity.At the end we use the predict-row function to predict the values.

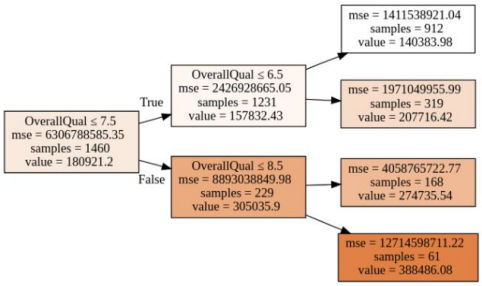

Figure 5. Decision Tree Model

k-fold Cross Validation: Cross-validation is a resampling procedure used to evaluate machine learning models on a limited data sample. The general procedure is as follows:

1.Shuffle the dataset randomly.
2.Split the dataset into k groups
3.For each unique group:
4.Take the group as a hold out or test data set
5.Take the remaining groups as a training data set
6.Fit a model on the training set and evaluate it on the test set
7.Retain the evaluation score and discard the model
8.Summarize the skill of the model using the sample of model evaluation scores

We use a four fold cross validation to our evaluate our model and its accuracy with F1-score.The results are shown for catalogue 1 in Table 1 in Results section.

Results

The results obtained using k-fold cross validation(4-folds) on the catalogue 1 are summarized in Table 1 below.

Fold	Accuracy(%)	F1-Score(%)	Precision(%)
1	93.78%	98.05%	96.17%
2	96.29%	99.36%	98.74%
3	93.82%	99.02%	98.06%
4	93.67%	99.35%	98.70%
Avg	94.39%	98.95%	97.92%

Table 1. Accuracy, F1-Score and Precision obtained for each fold

The results shown in the below Table2 summarizes about the accuracy achieved on each catalogue by our model.

Catalogue	Accuracy(%)
1	93.07%
2	91.82%
3	96.97%
4	86.51%

Table 2. Accuracy obtained on each catalogue

Average Class level Accuracy for Quasars was observed to be 97.17%

The model takes about 1 second to predict and about 30 minutes to train on a system with an i7-7500U processor, and 8GB RAM.

Other Models Used

Gaussian Mixture Models

Datasets can be modeled by Gaussian Distribution (Univariate or Multivariate). So it is quite natural and intuitive to assume that the clusters come from different Gaussian Distributions. Or in other words, it is tried to model the dataset as a mixture of several Gaussian Distributions. This is the core idea of this model.
In one dimension the probability density function of a Gaussian Distribution is given by
$G(X|\mu,\sigma) = \frac{1}{\sigma\sqrt{2\pi}} e^{-(x-\mu)^2/2\sigma^2}$ *where* μ and σ^2 are respectively mean and variance of the distribution.
For Multivariate (let us say d-variate) Gaussian Distribution, the probability density function is given by
$G(X|\mu,\Sigma) = \frac{1}{\sqrt{(2\pi)|\Sigma|}} \exp\left(-\frac{1}{2}(X-\mu)^T\Sigma^{-1}(X-\mu)\right)$ *Here* μ is a d dimensional vector denoting the mean of the distribution and Σ is the d X d covariance matrix.
Results:
The accuracy obtained was not really good in catalogue 3 which was equal to 75.75%.
The accuraacy is too low since the standard Deviation of the datapoints across the features of datset is much high.

Conclusion

Hence in this report we would like to conclude that Decision trees are a good model to binary classify the given dataset into Stars and Quasars, since the accuracy obtained are good values.

References

1. Machine Learning in Astronomy: A Case Study in Quasar-Star Classification
Authors: Mohammed Viquar, Suryoday Basak, Ariruna Dasgupta, Surbhi Agrawal, and Snehanshu Saha
2. Build a Decision Tree from Scratch in Python
Author: Venelin Valkov
3.**Github Link to our project:** Project Link