Machine Learning Project : Classification of Stars and Quasars Using Decision Trees with Bagging

Harshith U
PES1201700096

Ziyan Zafar
PES1201701910

Mukund Kadlabal
PES1201700648

Section 5'G' , PES UNIVERSITY

Abstract :

Through this project, we aim to apply the principles of Machine Learning in the field of Astronomy, and come up with a classification algorithm for differentiating Stars and Quasars - two types of astronomical bodies present abundantly in the universe. Just based on photometric data, these can be hard to distinguish between, and hence we are trying to implement machine learning models to classify them.

It is challenging to distinguish between a star and a quasar just based on photo metric surveys - they both have a compact optical morphology. Hence, we generally require spectroscopic data such as their optical variability or their optical colors to make the distinction. The ideal method is to just use the spectroscopic data so that red-shift can be used to differentiate betweenthe two. However, spectroscopic data is not always available and hence the need to implement machine learning models arise.

Problem Statement :

Provided with four catalogues, each with a collection of photometric data related to stars or quasars. We were required to classify each of the data points in the csv file of every catalog, as star or quasar. An accuracy of > 92% was expected using any machine learning model of our choice.

Dataset :

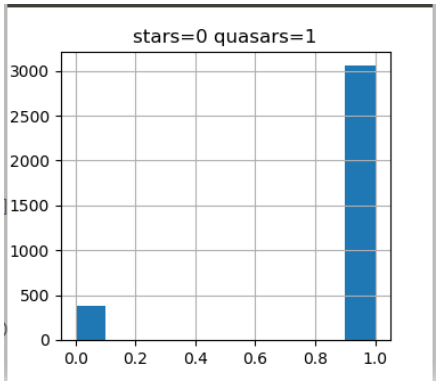
We are using only optical photo metric data and UV data, which was observed using the eSloan Digital Sky Survey (SDSS) and the Galaxy Evolution Explorer (GALEX) telescope respectively.
The data is from two regions, North Galactic Region and the Equatorial Region .
The 4 catalogues are a combination of data from the different regions , with different features.

Our Approach :

We first performed Exploratory data analysis to infer our dataset .
Tests for missing data , presence of outliers, tests for skewness and normality were performed , using histograms and density plots.

During EDA an interesting oddity was observed in catalogue 3 – The large difference in representation of each class star(0) and quasars(1) in the dataset.

To combat this difference in representation we have used an oversampling technique called **SMOTE** (Synthetic Minority Over-sampling Technique). In simple terms, it looks at the feature space for the minority class data points and considers its *k* nearest neighbours.



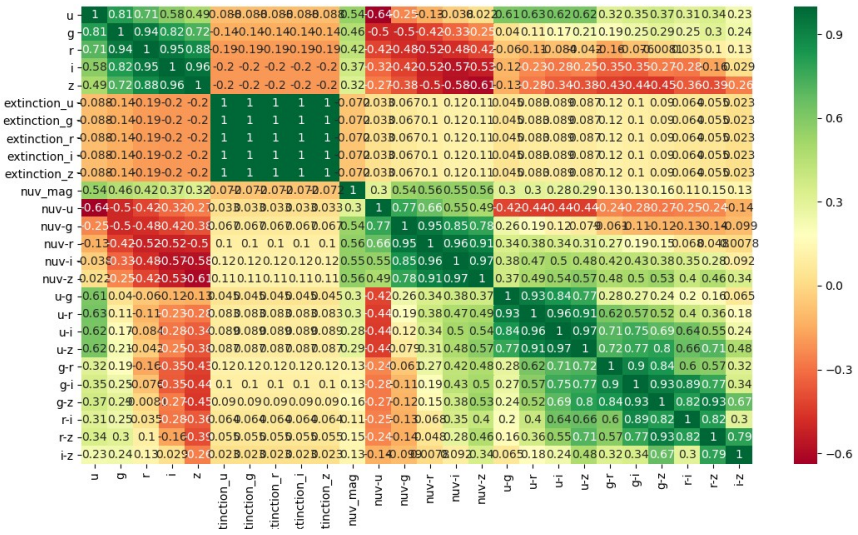
We tested the effectiveness of the oversampling technique by plotting a correlation matrix between all the features before and after applying Smote and the heatmap obtained after showed more higher correlation between features.

Feature Selection and Transformation :

Feature selection and transformation are important steps that which hugely impacts the performance of a machine learning model.We have used 2 methods correlation matrix with heatmap and a feature importance using extra tree classifier.

Correlation states how the features are related to each other or to the target variable.

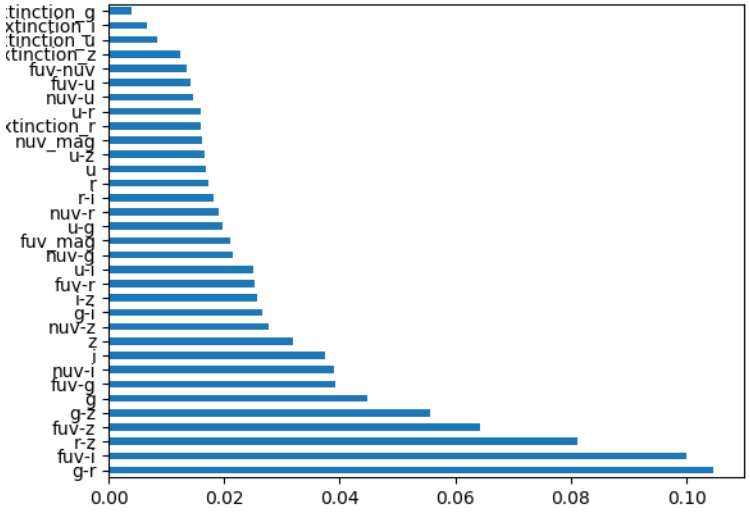
Correlation can be positive (increase in one value of feature increases the value of the target variable) or negative (increase in one value of feature decreases the value of the target variable)



Feature importance using extra tree classifier

Feature importance gives you a score for each feature of your data, the higher the score the more important or relevant the feature is towards your output variable.

Using the feature importance values we keep a threshold of 0.3 . Keeping all features with importance values above 0.3 and dropping the rest.



Algorithm Used : Classification and Regression tree

A decision tree classifier is a binary tree where predictions are made by traversing the tree from root to leaf — at each node, if a feature is less than a threshold , right otherwise.

Decision trees use the concept of **Gini impurity** to describe how "pure" a node is. A node is pure (*G = 0*) if all its samples belong to the same class, while a node with many samples from many different classes will have a Gini closer to 1.

$$G = 1 - \sum_{k=1}^{n} {p_k}^2$$

The training algorithm is a recursive algorithm called **CART**, short for *Classification And Regression Trees*. Each node is split so that the Gini impurity of the children (more specifically the average of the Gini of the children weighted by their size) is minimized.

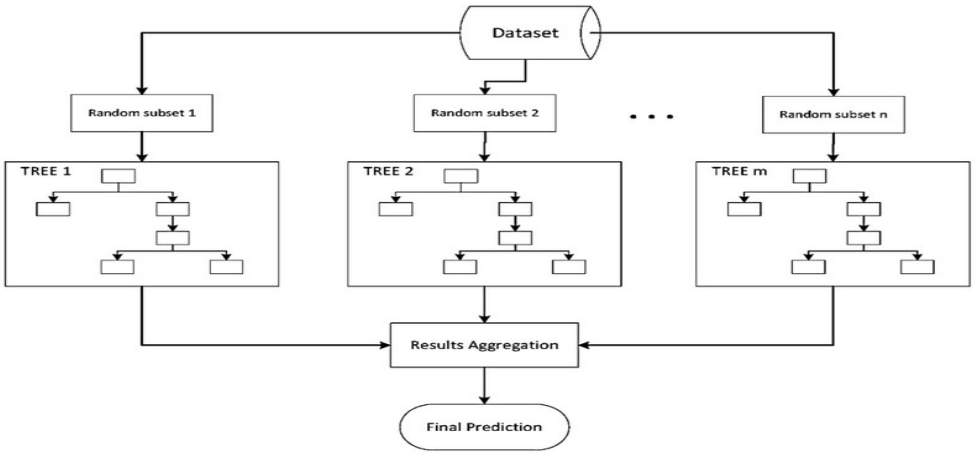
The recursion stops when the maximum depth is reached, or when no split can lead to two children purer than their parent.

The key to the CART algorithm is finding the optimal feature and threshold such that the Gini impurity is minimized.

$$G_i = \frac{i}{m} G_i^{left} + \frac{m-i}{m} G_i^{right}.$$

To improve the performance of our model we use the ensemble technique Bagging.

Bagging or Bootstrap Aggregation create multiple bootstrap subsamples from the data. A Decision Tree is formed on each of the bootstrapped subsamples. After each subsample Decision Tree has been formed, an algorithm is used to aggregate over the Decision Trees to form the most efficient predictor.



Hyperparameter tuning was done through which the hyperparameter max_depth was optimized to 7 for which our model gave best results.

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 10 fold cross validation to evaluate our model and its cross val accuracy The results are shown for catalogue 1.

```
k fold for decision trees

Scores: [0.90566038 0.98113208 0.9245283  1.         0.98076923 0.98076923
 0.96078431 1.         0.96078431 0.98039216]


Mean cross val accuracy: 0.9674820000569169
Standard Deviation: 0.029287522210194896
```

Catalogue	Accuracy
Catalogue 1	97%
Catalogue 2	94%
Catalogue 3	96%
Catalogue 4	86%

Cross validation Results

We further use the predicted values to calculate precision , recall , f1 score and support using the original class labels in testing data.

```
confusion matrix for decision trees

[[  9   2]
 [  1 118]]
Accuracy Score : 0.9769230769230769
Report :
              precision    recall  f1-score   support

           0       0.90      0.82      0.86        11
           1       0.98      0.99      0.99       119

    accuracy                           0.98       130
   macro avg       0.94      0.90      0.92       130
weighted avg       0.98      0.98      0.98       130
```

Summary for catalogue 1

Similarly for the other catalogues also high performance metrics were obtained .

Conclusion :

We use Decision Tree, specifically the Classification and Regression Trees algorithm for training our model, along with the ensemble method of Bootstrap aggregation or bagging and achieve an average accuracy of about 92 percent across all our catalogues

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 - Joachime valente
3.Ensemble methods for decision trees – Evan lutins