CLASSIFYING STARS AND QUASARS UE17CS303 ML ASSIGNMENT

https://github.com/shreyasmav/Classifying-Stars-and-Quasars-using-Decision-Tree

SHREYAS MAVANOOR
PES1201700837 Class 5E
CSE Department
PES University
Bangalore, India
shreyasmavanoor@gmail.com

SAQLAIN PASHA
PES1201701539 Class 5E
CSE Department
PES University
Bangalore, India
saqlainpasha1921@gmail.com

FAIZAN SIDDIQUI
PES1201701740 Class 5E
CSE Department
PES University
Bangalore, India
faizansbrsiddiqui@gmail.com

***Abstract*—One of the main problems in the world of astrophysics is differentiating between heavenly bodies such as stars and quasars which are distinct from each other. The main problem between scientists and researchers alike is that they have a hard time differentiating the two separate entities from the collected SDSS catalog data that comprises of the features and classification data. The current methods have proved to be inefficient and not up to par with modern expectations. Matching the recorded findings from the Galex is a long and rather tedious process. The decision tree that has been implemented is used to tackle this problem and achieve a satisfactory F1 score and accuracy.**

***Index Terms*—Keywords: Decision tree algorithm, Imbalanced Learning, Synthetic Minority Oversampling Technique, Sloan Digital Sky Survey, Galaxy Evolution Explorer**

I. Problem Statement

The Galaxy Evolution Explorer or GALEX for short is a space telescope that was used for the NASA Explorer program. It recorded astronomical sources in the far-UV and near-UV wavebands. The Sloan Digital Sky Survey or SDSS for short is a survey that observed a large portion of the sky in the following wave bands - u,g,r,i,z and then obtained the spectrum of the sources so that their red-shifts could be determined as well.

This project attempts is to classify photo-metric data collected from the GALEX and the SDSS over both the North Galactic region and Equatorial region in to spectroscopic classes of Stars and Quasars. Decision Tree Machine Learning algorithm is used to successfully distinguish between Stars and Quasars. Inferences about the data set and the results of the Decision tree model have been elucidated.

II. Machine Learning Techniques used

A. Context

Machine learning uses algorithms and statistical models that computer systems use to perform a specific task without using explicit instructions, relying on patterns and inference instead. Machine learning techniques are broadly classified into 2 types - Supervised and Unsupervised:

- Supervised learning: In this type of learning, the model knows what is it predicting. There are labels for all the samples. The model improves itself by comparing its prediction with the actual output.
- Unsupervised learning: in this type of learning, the model does not know what it is predicting. The model learns pattern from the given data. It makes groups with in data. For a new sample it compares it all the groups and adds it to the group with the most similarity.

B. Decision Tree Algorithm

Decision tree is a machine learning algorithm which is used for classification and regression. Decision tree algorithm keeps on dividing the dataset into smaller and smaller segments until each segment belongs to a single class. It can be viewed as if and else conditions. At the end it resembles a tree with nodes and leaves. At each node the data is split based on some condition. At each of the leaves, data will belong to a single class. The data at each node will be split based on some pre-determined metric. For a new data sample, it divides that sample based on pre-learned set of rules. At the end, the algorithm returns the class of the node to which the samples goes.

C. Synthetic Minority Oversampling Technique

SMOTE stands for Synthetic Minority Oversampling Technique. Machine learning algorithms cannot perform well when one class dominates other. SMOTE is an oversampling technique, which generates synthetic samples from the minority class to balance to dataset . It generates samples by considering the existing ones. It takes one sample form the minority class at a time replaces value of one the feature at a time by taking the average of that sample and its neighboring sample .

III. Detailed Methodology

A. Pre-processing the data

Catalog 3 which had data from both the regions (North Galactic Pole and Equatorial Region) and only samples which

had fuv values were used. '$Galex\_objid$','$SDSS\_objid$', '$Pred$','$class$'. Also, '$spectrometric\_redshift$' columns are dropped while training the model. As the data was imbalanced, we used SMOTE to balance both the classes.

B. Selecting a classification model

The Random Forest Algorithm is by far the most efficient and is the first choice made by most researcher to tackle this unique problem as the trees are more diverse and can handle over-fitting better. Also, a decision tree is nothing but a tiny subset of a Random Forest.

A decision tree classification model is a b-tree where the predictions are made by traversing the tree from the root to the leaf where at each node, we can go left if a feature $<$ threshold else take a right. Lastly, each leaf present here is associated with a class, which is the output of the predictor.

To split the data at each node we use a measure called Gini, which is an impurity that describes how pure a node is. When $G = 0$ at a node, it is pure which means that all the samples belong to the same class. A node containing many different samples from different classes will have a Gini that tends to be closer to 1.

The Gini impurity of n training samples divided across k classes can be defined as

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

Fig. 1. where p[k] = fraction of samples within class k

The training algorithm is a recursive algorithm called CART also known as Classification and Regression Trees. CART is a decision tree algorithm which uses gini index as the metric to split the data at each node. CART is used for both classification and regression tasks. Gini index stores the sum of probabilities for each class. At each index the algorithm calculates all possible gini index values and chooses the best one.

If m is size of the node and m[k] the number of samples of class k in the node, and then after observing the i-th threshold

$$G = 1 - \sum_{k=1}^{n} {p_k}^2 = 1 - \sum_{k=1}^{n} \left(\frac{m_k}{m}\right)^2$$

we can see i elements on the left and m–i elements on the right, and The final Gini expression is a simple weighted average:

$$G_i^{left} = 1 - \sum_{k=1}^{n} \left(\frac{m_k^{left}}{i}\right)^2$$

$$G_i^{right} = 1 - \sum_{k=1}^{n} \left(\frac{m_k^{right}}{m-i}\right)^2 .$$

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

Fig. 2. Simple weighted average(Gini)

C. Training and Testing the Classification model

The dataset has been split in 70:30 train-test ratio using $sklearn.train\_test\_split()$. The accuracy of the model is compared with and without using SMOTE. A score method has been implemented in the DecisionTreeClassifier class to get the accuracy. The model was trained on a range of values for max_depth parameter to get the most optimal value for max_depth.

IV. Result and Conclusion

After training and testing the Decision Tree Classifier, it was found out that the optimal value of max_depth was 8 and sometimes 9. The following are the results for when max_depth was 8. A graph of accuracy for different

Without SMOTE				With SMOTE			
Class	Precision	Recall	F1-score	class	Precision	Recall	F1-score
0	0.83	0.72	0.77	0	0.64	0.79	0.71
1	0.97	0.98	0.97	1	0.97	0.94	0.96
Accuracy = 0.95				Accuracy = 0.93			

Fig. 3. Table shows tree with smote vs without smote

max_depth values is plotted. The ROC curve and the PR curve are also plotted.

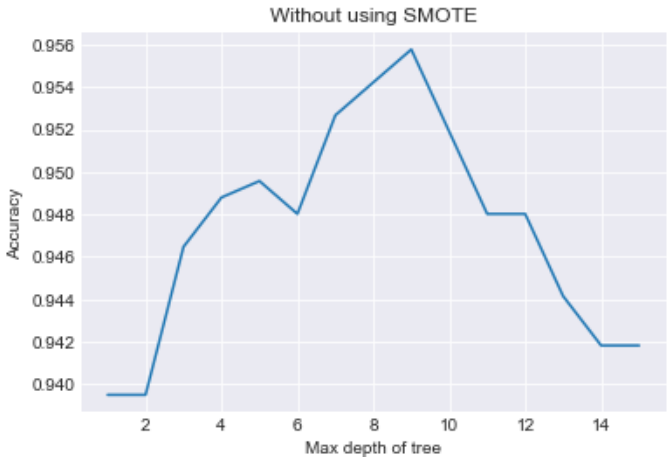

Fig. 4. Decision tree without SMOTE

The following are the results while using different set of features for a value of 8 for max_depth hyper-parameter without using SMOTE:

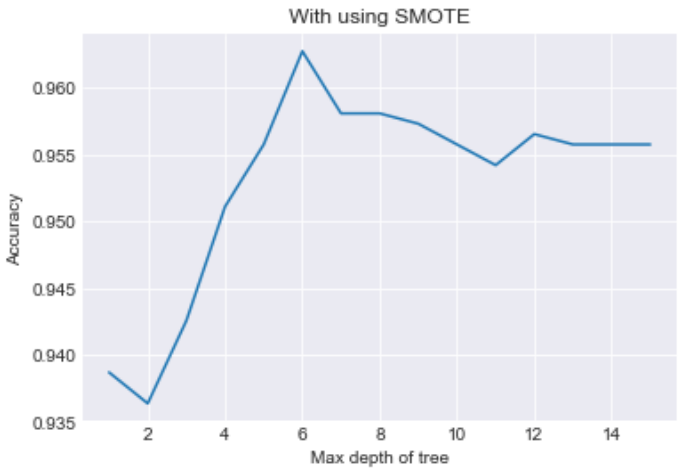

Fig. 5. Decision tree with SMOTE

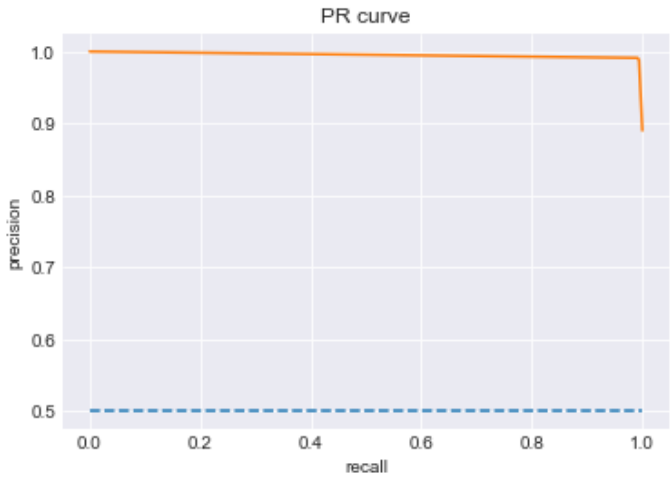

Fig. 6. Graph showing PR curve

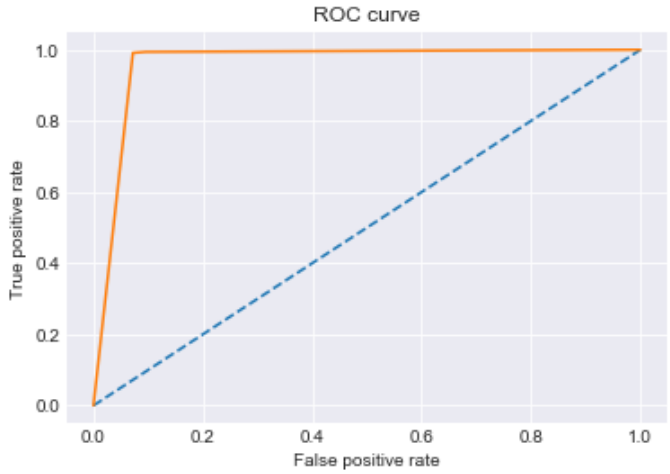

Fig. 7. Graph showing ROC curve

- All features are used - 0.95 Accuracy
- Extinction values are dropped - 0.96 Accuracy
- Pair wise differences are dropped - 0.93 Accuracy
- Extinction and pair wise differences are dropped - 0.93 Accuracy

Cross Validation

Cross validation using spectrometric redshift.

1) Range 1: $z <= 0 : 0033$: We expect the types of samples in this range to be predominantly stars
2) Range 2: $z >= 0 : 004$: We expect the types of samples in this range to be predominantly quasars
3) Range 3: $0 : 0033 < z < 0 : 004$: This range of red shifts represents the overlap

We are getting an accuracy of 0.64 while cross validating the model based on spectrometric redshift.

			Predicted class	
Catalog	Range	True Class	Star	Quasar
3	1	Star Quasar	415 8	37 0
3	2	Star Quasar	10 0	1 0
3	3	Star Quasar	8 121	0 3695

Fig. 8. Cross Validation table

REFERENCES

[1] Makhija, Simran & Saha, Snehanshu & Das, Mousumi & Basak, Suryoday. (2019). Separating Stars from Quasars: Machine Learning Investigation Using Photometric Data. 10.13140/RG.2.2.24220.74889.
[2] https://medium.com/greyatom/decision-trees-a-simple-way-to-visualize-a-decision-dc506a403aeb