Classification of Stars and Quasars using K-nearest neighbours *

Srivatsa Vantmuri
PES1201701132
srivatsavantmuri@gmail.com

Shashidhar R
PES1201701417
shashir14399@gmail.com

Saransh Gupta
PES1201700267
saranshgupta2407@gmail.com

***Abstract*—Using Machine Learning to classify stars and quasars from data obtained from Galex and SDSS photometric data. We used the K-nearest neighbors approach do this task. We measured the correctness for this by accuracy and other parameters such as precision, recall and F1 score.**

I. Introduction

Stars and quasars are celestial objects which have very little visual difference when observed from our planet.A star consists of a luminous spheroid of plasma held together by its own gravity.A quasar is an extremely luminous active galactic nucleus in which a supermassive black hole is surrounded by a gaseous accretion disk. They seem to be indistinguishable when merely observed however they differ in many aspects, especially their photometric features. Stars and quasars look very similar in their optical images but the spectral energy distribution for stars and quasars is different and so the optical bands from SDSS namely u, g, r, i and z can be used to separate them. The vast differentiating factor is their UV emissions. In this project both optical and ultraviolet (UV) photometric data is used with machine learning methods (KNN) to discriminate between stars and quasars. The spectroscopic labels are used as the primary class label. Both stars and quasars have a compact optical morphology and are hence difficult to separate without spectroscopic data. In such cases, other parameters of the sources such as their optical variability or their optical colors are necessary to distinguish between them. Further research has shown that including the infrared data or UV data with optical photometry results in a more efficient separation. Using data from GALEX, cross-matching with labels in SDSS, provided data from both the visible and ultraviolet spectra. Each of the photometric samples chosen in either region (north galactic or equatorial) have an associated spectroscopic label from the SDSS database.

II. Machine Learning Approach

After loading the dataset we stored the column **'spectrometric_redshift' galex_objid** and **sdss_objid** in a list.
After that we dropped it from the dataset as we don't need it until verification step. We then dropped **'pred'** column as we don't train our model based on that. We split the Data Set in in 70:30 ratio and store it in Training Set and Test Set and fit model on Training Set. We Then UpSample the Training Set. We used KNN Based Algorithm to solve the problem. We Initially pre-process the data using sklearn by standardizing all the column values.

z = (X - Mu)/sigma

X: the observation (a specific value that you are calculating the z-score for).
Mu: the mean.
sigma : the standard deviation
The class value for majority_class would be maxixmum value of "class" column.
the class value for minprity_class would be minimum value of "class" column.
We used Manhattan Distance to check for nearest neighbour condition.
Formula : distance = summation(xi - yi)
i is total total attributes of each instance.
We then used K = 3 ((neighbours) which is optimal value) and called **getNeighs()** function to get neigbours for each instance of Training Set
We then call **getResponse()** to get votes for nearest neighbours and predict the class .

A. Upsampling

We basically UpSample Training Set to make class rows equal in number. We separated the Stars and Quasars rows and store them under list majority_class and minority_class respectively. We find out the difference in number of rows in majority_class and minority_class and store the difference in n_samples . We make copies of minority class rows and append them in the same list at the end until both list become equal in length . We then concatenate both list and convert it into a Dataframe and store it in Training Set.

Input : UpSampling(TrainngSet).
Output : UpSampled DataSet.

```
for each row in TrainingSet do
  if row belongs to class 1
    majority_class.append row
  else
    minority_class.append row

nsamples = len(majority_class)-len(minority_class)
unsampled = []
```

for i in range(range(nsamples)) **do**
  j = 0
  if j ¡ nsamples
    unsampled.append minority_class[j]
  else
    j = 0
    unsampled.append minority_class[j]

concat(minority_class , unsampled)
concat(majority_class , minority_class)
DataSet DataFrame(majority_class)
return DataSet

III. Results

We call **getAccuracy()** function to find **precesion**, **recall** , **f1 score** for each class.

We append each class value of Test Set into a list. We then compare each Test Set **"class"** value with predicted values and find the **True Positive(TP)** which is correctly classified quasars,**True(TN) Negative** which is correctly classified stars,**False Negative(FN)** which is incorrectly classified stars,**False Postive(FP)** which is incorrectly classified quasars values. Using this we calculated overall accuracy and class wise precision, recall and f1 score using the below formulae
Accuracy = (TP+TN)/(TP+TN+FP+FN)
For class stars
precision = TN/(TN+FN)
recall = TN/(TN+FP)
f1 score = 2*precision*recall/(precision+recall)
For class Quasars
precision = TP/(TP+FP)
recall = TP/(TP+FN)
f1 score = 2*precision*recall/(precision+recall)

We then plot ROC-Curve by using sklearn.
We inintially get values of **TPR** , **FPR**
We then calculate **auc** value . We then plot FPR , TPR along x-axis and y-axis respectively

Following are the some of the results from four catalogs ROC-Curve For all catalogs.

Accuracy Measures for Stars				
Calalog	Accuracy	Precision	Recall	F1 score
Calalog-1	0.97	0.98	0.72	0.76
Calalog-2	0.97	0.90	0.74	0.81
Calalog-3	0.95	0.94	0.73	0.80
Calalog-4	0.86	0.75	0.78	0.77
Accuracy Measures for Quasars				
Calalog	Accuracy	Precision	Recall	F1 score
Calalog-1	0.97	0.96	0.98	0.98
Calalog-2	0.97	0.97	0.99	0.98
Calalog-3	0.95	0.95	0.99	0.97
Calalog-4	0.86	0.91	0.89	0.90

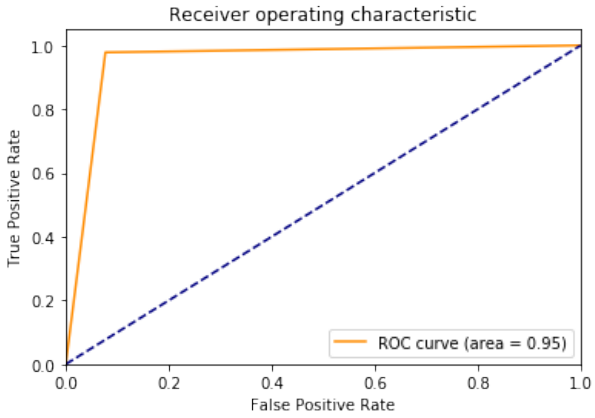

Fig. 1. cat1.csv

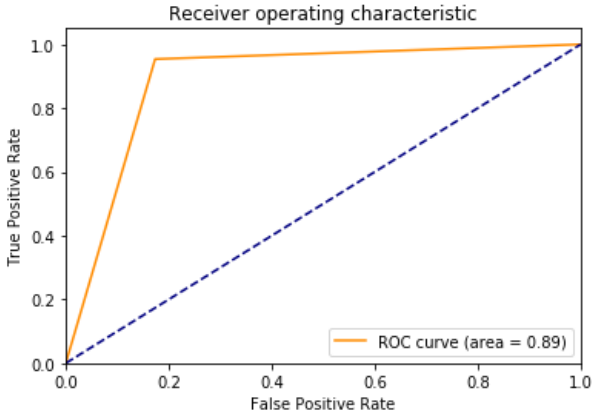

Fig. 2. cat2.csv

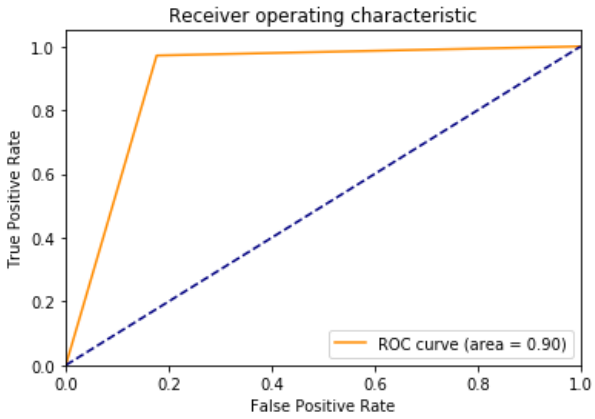

Fig. 3. cat3.csv

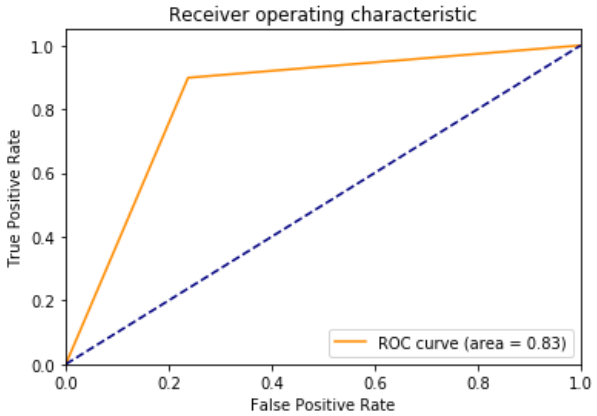

Fig. 4. cat4.csv

IV. Conclusion

After running KNN Algorithm on the Data Set , the accuracy obtained was greater than 95% for first 3 catalogs and 86% on cat4.csv catalog with K = 3.
We plotted graph of K-value vs accuracy and found out optimum K-value.

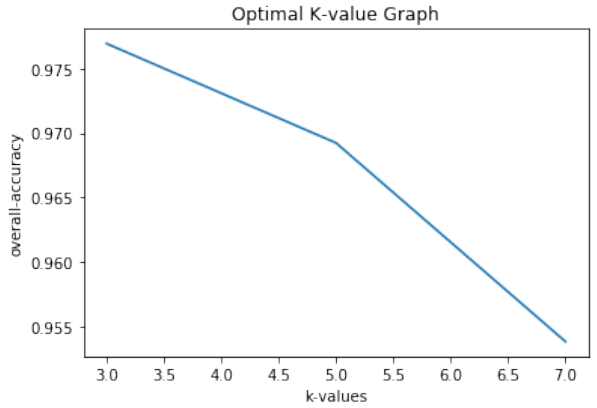

Fig. 5. cat4.csv

V. References

[1] Simran Makhijaa, Snehanshu Saha, Suryoday Basak,Mousumi Dasd. Separating Stars from Quasars: Machine Learning Investigation Using Photometric Data.