UE17CS303-MACHINE LEARNING

PES1201700118 Saioni Chatterjee
PES1201700770 Aishwarya Pramod
PES1201701583 Aprameya Kulkarni

November 2019

1 Introduction

One of the major challenges of large scale photometric surveys is the separation of the different classes of sources, especially stars and quasars.Both types of sources have a compact optical morphology and are hence difficult to separate without spectroscopic data.

In this paper we present one machine learning classification approaches to distinguish between stars and quasars using only optical photometric data and UV data, observed using SDSS(Sloan Digital Sky Survey) and GALEX(Galaxy Evolution Explorer) respectively.

Machine learning involves the use of statistics to make useful predictions and to learn essential features; classification is the process of marking separations between categories in data.

The best method is to include spectroscopic data so that the red-shift of quasars, can be used to differentiate between stars without our Galaxy and quasars.

2 Data and Problem Statement

GALEX is a space telescope, which observed astronomical sources in the far-UV and near-UV (FUV and NUV) wavebands. The Sloan Digital Sky Survey or SDSS is an optical survey that observed large portions of the sky in the wave bands u,g,r,i,z and obtained the spectra of the sources so that their red-shifts could be determined as well.
We have taken into consideration the GALEX and SDSS matches from two regions.
1. North Galactic Region: Data from the region ¿ 75 of galactic latitude is used. Since UV emission is significantly affected by dust extinction, the reason to consider the data from this region is that the stellar extinction is constant and low. From this region, we had used 2027 quasars and 912 stars.

2. Equatorial Region: The extinction observed around the equator is varied and poses additional challenges to the classification process. We have selected data in the range of -30 dec to 30 dec and have used the extinction values present in the SDSS catalog to account for the extinction. The data from this region had 9182 stars and 20716 quasars.
In our experiments, we use the spectroscopic label as the primary class labels.Thus, the problem that we're trying to primarily address is to classify photometric data into spectroscopic classes.

3 Naive Bayes

(ML technique Employed with some detail)

Naive Bayes is a very powerful algorithm used for prediction as well as classification. It follows the principle of "Conditional Probability,i.e. Bayes theorem.

The algorithm is called Naive because it assumes that the features in a class are unrelated to the other features and all of them independently contribute to the probability calculation.For instance if you take features like temperature, humidity and wind speed to predict the rain you would assume that all those three features independently contribute to probability of upcoming rain. Even if these features have some relation we would naively tell that they are not.Naive Bayes algorithm is very handy on very large data sets, because it's fast, simple and accurate when compared to other classification algorithms.

Bayes Theorem describes the probability of an event based upon some prior knowledge related to that event. In layman's terms, it gives the probability of an event A, given another event B has already occurred.

P(A—B) = (P(B—A).P(A)) / P(B)

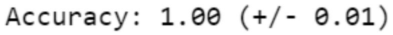
-P(A/B) – Probability of occurrence of event A, given event B, has already occurred.
-P(A) – Probability of occurrence of event A.
-P(B) – Probability of occurrence of event B.
-P(B/A) – Probability of occurrence of event B, given event A has already occurred.

In a Bayes classifier, we calculate the posterior (technically we only calculate the numerator of the posterior) for every class for each observation. Then, classify the observation based on the class with the largest posterior value.

4 Summary of Results

Outcome using Naive Bayes Classifier

The results obtained use 3 fold cross-validation, where 3 disjoint test sets are used to test the model, after training on the remaining data, in order to derive a more accurate estimate of model prediction and alleviate concerns of overfitting. This method of cross-validation also serves the purpose of the model accommodating statistical variance in new(unseen) test data.



```
Accuracy: 1.00 (+/- 0.01)
```

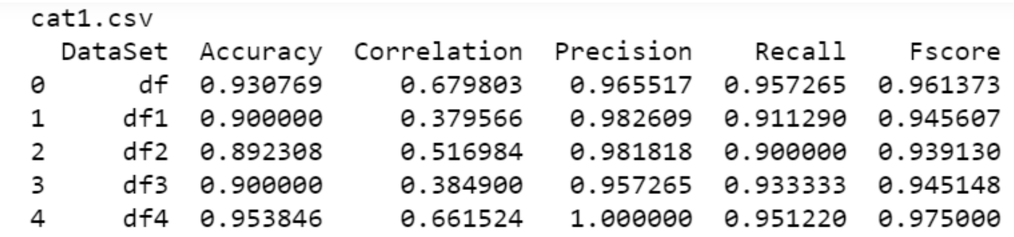
The different subset of features used are summarized into cases to determine the best fit based on computation power and size of the dataset:

case 1 : Parsimony list : Using fuv-nuv, nuv-r and r: The purpose of using these features is to determine the effectiveness of machine learning on a very small set of features that are thought to be effective in discriminating between stars and quasars

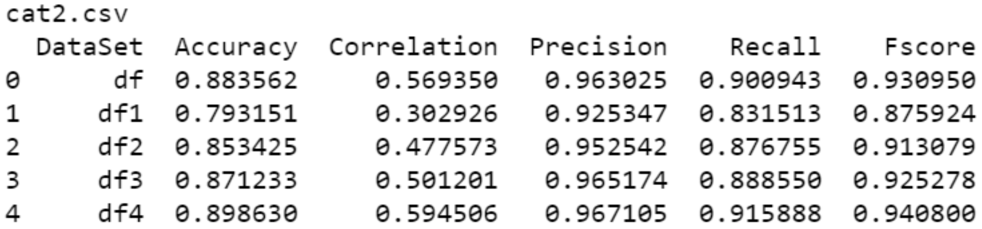
Case 2 : Dropped nuv and pairwise difference between u, g, r, i ,z.Here the near ultra violet attribute is excluded from the data and comparatively the accuracy decreased.

Case 3 : Dropped-fuv list : Using nuv, u; g; r; i; z and pairwise difference between them: One of the reasons to exclude the fuv attribute from the data is that the far-ultraviolet detector was damaged in April 2010 and subsequent observations were done only using the near-ultraviolet detector. Hence, from a data-analysis point of view, this poses an important challenge as a lot of fuv observations are not reliable.

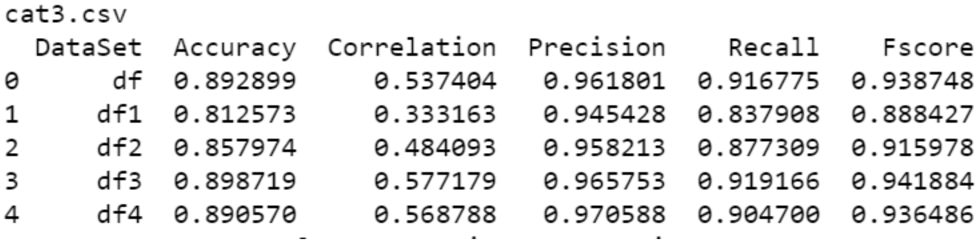
Case 4 : Existing-fuv list : Using existing fuv; nuv; u; g; r; i; z and pairwise difference between them: Here we select all the data that do not have any missing values of fuv. The purpose of this case is to estimate how well the classifiers can perform when all the features are present.



```
cat1.csv
  DataSet  Accuracy  Correlation  Precision    Recall    Fscore
0      df  0.930769     0.679803   0.965517  0.957265  0.961373
1     df1  0.900000     0.379566   0.982609  0.911290  0.945607
2     df2  0.892308     0.516984   0.981818  0.900000  0.939130
3     df3  0.900000     0.384900   0.957265  0.933333  0.945148
4     df4  0.953846     0.661524   1.000000  0.951220  0.975000
```



```
cat2.csv
  DataSet  Accuracy  Correlation  Precision    Recall    Fscore
0      df  0.883562     0.569350   0.963025  0.900943  0.930950
1     df1  0.793151     0.302926   0.925347  0.831513  0.875924
2     df2  0.853425     0.477573   0.952542  0.876755  0.913079
3     df3  0.871233     0.501201   0.965174  0.888550  0.925278
4     df4  0.898630     0.594506   0.967105  0.915888  0.940800
```

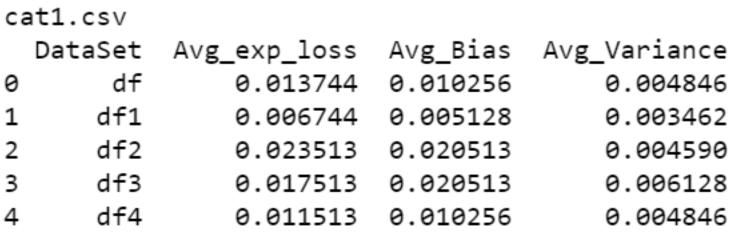


```
cat3.csv
  DataSet  Accuracy  Correlation  Precision    Recall    Fscore
0      df  0.892899     0.537404   0.961801  0.916775  0.938748
1     df1  0.812573     0.333163   0.945428  0.837908  0.888427
2     df2  0.857974     0.484093   0.958213  0.877309  0.915978
3     df3  0.898719     0.577179   0.965753  0.919166  0.941884
4     df4  0.890570     0.568788   0.970588  0.904700  0.936486
```

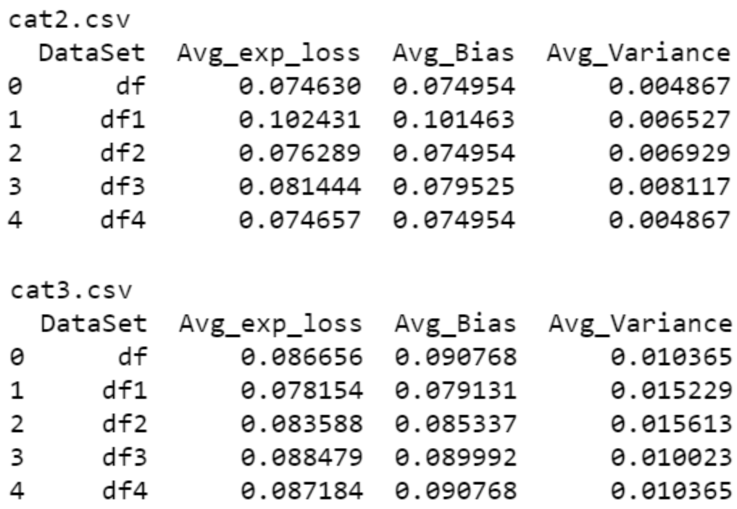
Bias-Variance Trade-off

The bias-variance decomposition describes the performance of the learning algorithm, as these concepts are correlated to underfitting and overfitting.
Bias is calculated as the difference between the expected value of the estimator and the parameter that we want to estimate. High Bias is proportional to underfitting.
Variance is calculated as the difference between the expected value of the squared estimator and squared expectation of the estimator.High Variance is proportional to overfitting.
Loss is calculated as the difference between bias and variance.
If a model has a very bias, increasing the variance can be beneficial as it would push the decision boundary, thus improving the loss.

Outcome using Bias-Variance trade off in Naive Bayes Model



```
cat1.csv
  DataSet  Avg_exp_loss  Avg_Bias  Avg_Variance
0      df      0.013744  0.010256      0.004846
1     df1      0.006744  0.005128      0.003462
2     df2      0.023513  0.020513      0.004590
3     df3      0.017513  0.020513      0.006128
4     df4      0.011513  0.010256      0.004846
```



```
cat2.csv
  DataSet  Avg_exp_loss  Avg_Bias  Avg_Variance
0      df      0.074630  0.074954      0.004867
1     df1      0.102431  0.101463      0.006527
2     df2      0.076289  0.074954      0.006929
3     df3      0.081444  0.079525      0.008117
4     df4      0.074657  0.074954      0.004867

cat3.csv
  DataSet  Avg_exp_loss  Avg_Bias  Avg_Variance
0      df      0.086656  0.090768      0.010365
1     df1      0.078154  0.079131      0.015229
2     df2      0.083588  0.085337      0.015613
3     df3      0.088479  0.089992      0.010023
4     df4      0.087184  0.090768      0.010365
```

5 Conclusion

The Naive Bayes has been successfully able to classify stars and quasars with relatively good accuracy. The reason for choosing Naive Bayes model is that its simple and if the conditional independence assumption actually holds, the classifier will converge quicker than discriminative models like logistic regression, so we need less training data. It requires less model training time.