COMPUTER SCIENCE AND ENGINEERING

Machine Learning
UE17CS303

Pavan Mitra	PES1201700239
Ambarish D Y	PES1201701635
Ravichandra G.	PES1201701581

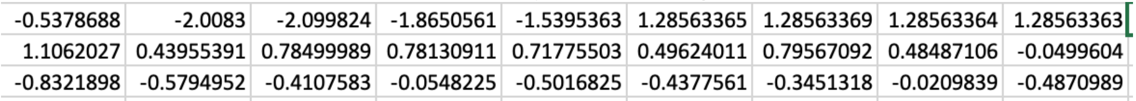
---

*STAR-QUASAR
CLASSIFICATION
Understanding data-model
metrics with a Naïve Bayes
model*

Contents

1 Why Bayesian

- We notice that the feature vectors comfortably come from nice little Gaussian distributions. We confirm this by calculating skews for each variable.

-0.5378688	-2.0083	-2.099824	-1.8650561	-1.5395363	1.28563365	1.28563369	1.28563364	1.28563363
1.1062027	0.43955391	0.78499989	0.78130911	0.71775503	0.49624011	0.79567092	0.48487106	-0.0499604
-0.8321898	-0.5794952	-0.4107583	-0.0548225	-0.5016825	-0.4377561	-0.3451318	-0.0209839	-0.4870989

  Ignoring the first 3 values(IDs) that are not features anyways, others have -2 >skews < 2, which is acceptable as symmetric.

- We observe that photometric band filter readings are independent variables, and hence are good candidates for a Bayesian model.

- We would like a simple model that generalises well over different sets of the same type/features.

2 Introduction

The Naïve Bayes model is built on the Bayes equation, combining prior knowledge with observed data

$$P(h|D) = \frac{P(D|h)P(h)}{P(D)}$$

$P(h)$ = prior probability of hypothesis h

$P(D)$ = prior probability of training data D

$P(h|D)$ = probability of h given D

$P(D|h)$ = probability of D given h

3 General Procedure

We follow a standard implementation of training procedures on catalogs 1 - 3:

- **Separate**: We separate the data based on class levels 0 and 1.
- **Summarize**:Find the feature summaries for each class level i.e Means and Standard Deviations.
- **Obtain feature-wise probabilities**: Compute them as

$$P_i(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-(x-\mu)^2 / 2\sigma^2} \tag{1}$$

- Obtain row-wise probabilities, using:

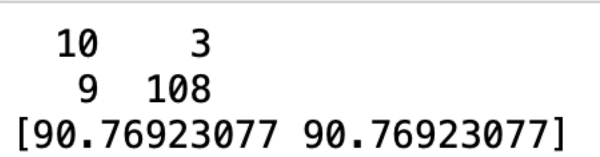
$$P_i(F|c_1c_2...c_n) = P_i(F|c_1).P_i(F|c_2).P_i(F|c_3)...P_i(F|c_n) \tag{2}$$

- Put the "Naïve" in Naïve Bayes, by using the assumption

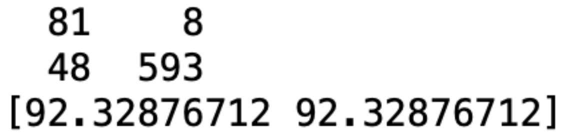
$$h_{ml} = \arg\max_{h_i \in H}(P(D|h_i)) \tag{3}$$

3.0.1 Report Results

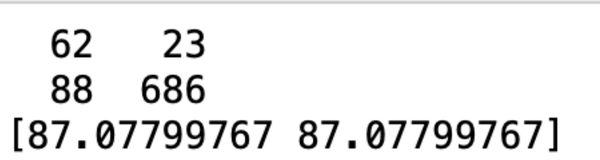
We report our confusion matrices and accuracy values. All matrices are shown for cat1,cat2,cat3,cat4 in order.



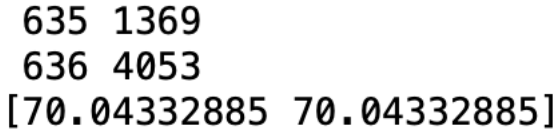
```
  10    3
   9  108
[90.76923077 90.76923077]
```



```
  81    8
  48  593
[92.32876712 92.32876712]
```



```
  62   23
  88  686
[87.07799767 87.07799767]
```



```
 635 1369
 636 4053
[70.04332885 70.04332885]
```

3.1 Improvements

- We look to include log probabilities over regular probabilities, since they work well at small values. We report the new readings.

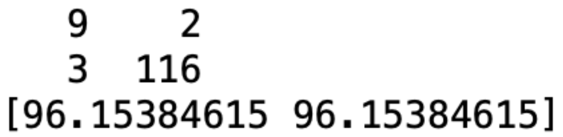
```python
    probabs[classValue]=0
        for i in range(len(classSummaries)):
            avrg,stdev=classSummaries[i]
            x=inputVector[i]
            probabs[classValue]+=math.log(abs(Prob(x,avrg,stdev))) #print(probabs)

```

Listing 1: Changing to log probs

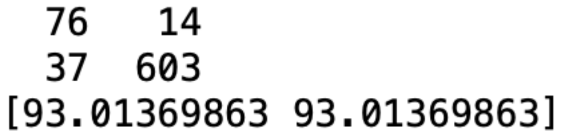
- We include prior probabilities as

$$P_i(h) = \frac{n_i}{\Sigma n_i} \tag{4}$$

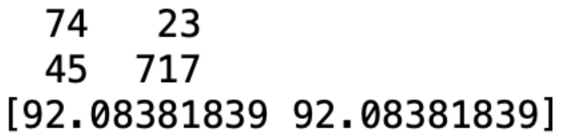
and report updated results



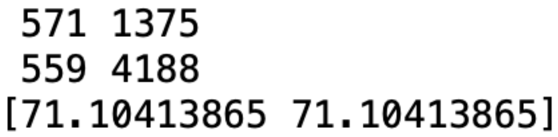
```
  9    2
  3  116
[96.15384615 96.15384615]
```



```
  76   14
  37  603
[93.01369863 93.01369863]
```



```
  74   23
  45  717
[92.08381839 92.08381839]
```



```
 571 1375
 559 4188
[71.10413865 71.10413865]
```

4 Special Case - catalog4

We observe that our model still performs poorly against catalog4, which has points from both *North Galactic* and *Equatorial* phases. We make the following observations:

4.1 Blindness

We calculate the ”blindness’ in data as fraction of points belonging to the majority class. This is because it can be interpreted as the accuracy that can be obtained if we predicted all points as belonging to one class.

4.2 Possible Solutions to the problem

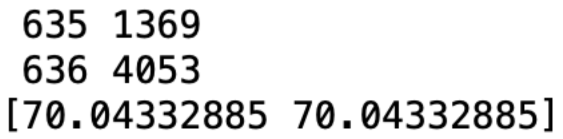
4.2.1 Missing fuv values

We try and impute the missing *fuv* values using *nuv* by means of a Linear Regression Model, as it has been previously observed that they highly correlate.

```
    model = LinearRegression().fit(x_, y2)
    r_sq = model.score(x_, y2)
    intercept, coefficients = model.intercept_, model.coef_
    print(intercept, coefficients)

```

Listing 2: Changing to log probs

We report the readings after including fuv readings.



```
 635 1369
 636 4053
[70.04332885 70.04332885]
```

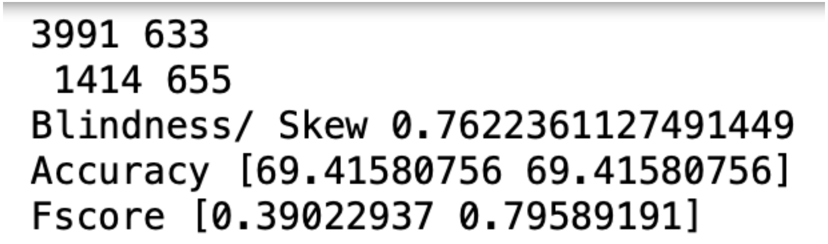
Again, this doesn't contribute much to out accuracy, it only re-adjusts the matrix entries by raising TN, FN values and lowers TP counts.

4.2.2 Is accuracy a good measure

We begin to doubt that our metric itself is not a good measure of the work our model is putting in, and hence we look to other possibilities. Since the data is skewed towards quasars and we are dealing with a high recall-low precision model, we look to F scores (Harmonic means of Precision an Recall).

4.2.3 Deviation from blindness

The difference bewtween accuracy and blindness



```
3991 633
 1414 655
Blindness/ Skew 0.7622361127491449
Accuracy [69.41580756 69.41580756]
Fscore [0.39022937 0.79589191]
```

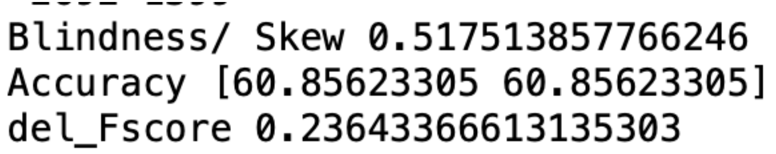
Further, we infer that F scores again are unable to tell the entire story (since they differ for both classes), and hence a contempt metric would be △ F, which denotes difference in F scores for each class.

5 Final Optimisations

Now that we have a metric in hand, we must tune our model to bring out results that optimise that metric.

5.1 Sampling

In order to eliminate the skew variations, we decide to implement an oversampling (of the minority class) technique (contrary to undersampling where parts of the majority class is axed). Changes are made and results are noted.

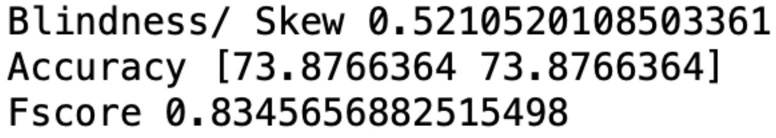


```
Blindness/ Skew 0.517513857766246
Accuracy [60.85623305 60.85623305]
del_Fscore 0.23643366613135303
```

Well well, our accuracy has fallen, but what is this we see, an improvement in both △ F scores, and deviation from blindness(-3 to +12).
The obvious question is, can we do better?

5.2 Inter-Model test-train

We find that a little bit of skew in the **test** data would be representative of our original population. Hence, we *train* our model on the newly normalised(upsampled) data, but test it against the original. Results are reported:



```
Blindness/ Skew 0.5210520108503361
Accuracy [73.8766364 73.8766364]
Fscore 0.8345656882515498
```

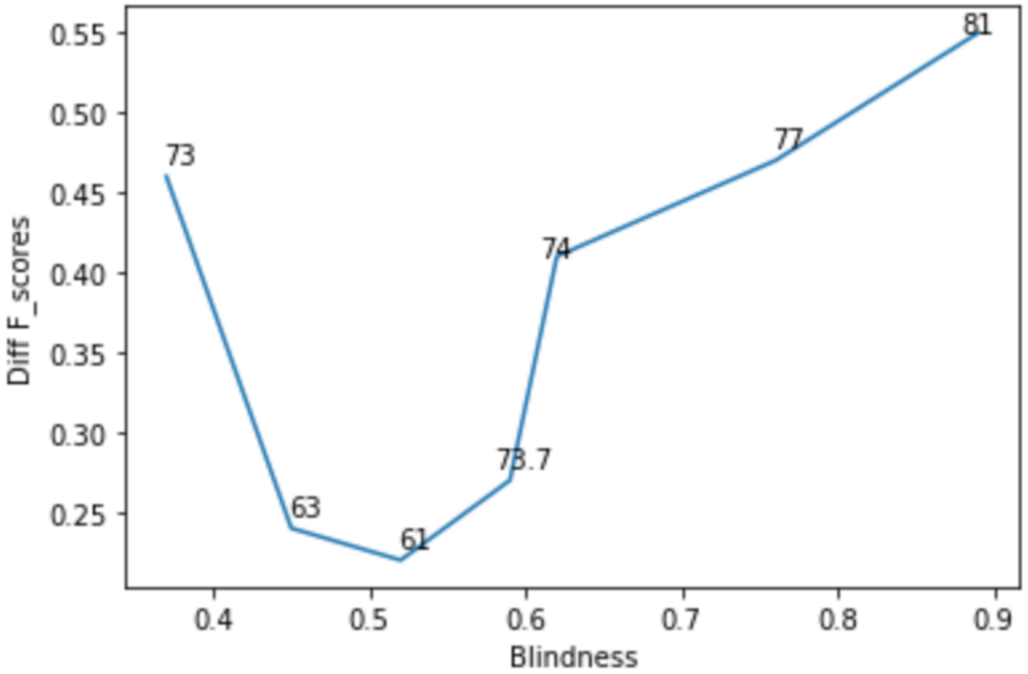
So we were able to optimise all 3 readings! This shows our model is doing well, but is there another aspect that we could manipulate?

5.3 Blindness in the oversampled sets

We realise that we could choose the blindness levels in the training set. Now, the variables that would change as a result of this are △ F score, and Accuracy. Namely, the former decreases and the latter increases as the blindness moves either way from the ideal 0.5 i.e. we now have an optimisation problem at hand where the objectives are to minimise △ F scores, maximise accuracy, and minimise (skew - 0.5)^2

$$S = \frac{Acc(M)^k}{(\triangle F)^m.(b - 0.5)^n} \tag{5}$$

We observe experimentally the following results:



From the figure, we report by visual inference that optimally, we would choose the model that brings blindness to around 0.58, which gives an accuracy of 73.7 and at a 0.26 F score.

6 Possible Improvements

- We could pick k, m and n and quantify the optimised solution we picked by visual observation.

- Automate the process of oversampling.
- Develop a heuristic to *pick* the samples to oversample.

7 Conclusion

We have developed a Naïve Bayes binary classification model, that sifts stars from quasars, which is measured against standard accuracy, against catalogs 1 through 3, and a custom score against catalog 4.