Magic Classification: Comparing Naive Bayes to SVMs

Bryce Elizabeth Yahn
University of Rochester
Rochester, New York
byahn2@u.rochester.edu

Jeremy Leonard Atkins
University of Rochester
Rochester, New York
jatkins5@u.rochester.edu

ABSTRACT

Classification tasks are one of the primary applications of data mining, and are found in a variety of different fields and industries. In this paper, we compare the performance of two classification methods on the Magic dataset. We perform basic data visualization and preprocessing, and then implement and compare the effectiveness of Gaussian Naive Bayes and Support Vector Machine models. The SVM model significantly outperformed the Naive Bayes, likely due to the dataset not sufficiently fulfilling Naive Bayes' independence assumption.

This analysis was conducted as the final project for CSC 240 at the University of Rochester in Fall of 2020, using material from Zaki and Meira's *Data Mining and Machine Learning* [1].

1 INTRODUCTION

Classification tasks are common data mining problems in a variety of fields. The goal of classification is to create a model that predicts the label or class for a given unlabeled point based that point's features. This is a supervised learning task which requires a training set of labeled points to create the model, which can then be used to make predictions for unlabeled points.

There are a number of different methods for classification which vary in their complexity including Bayesian classifiers, decision tree methods, support vector machines (SVMs) and neural networks, among others. Each of these approaches has their strengths and weaknesses, and which method is most effective depends on the properties of the dataset. While more advanced methods may be particularly powerful, they can overfit the if there are too many parameters relative to labeled datapoints.

One area of study in which these classification methods can be applied is astrophysics. For our comparison, we use the Magic Gamma Telescope (Magic) dataset from the UCI Machine Learning repository, generated by Bock et. al. for their paper *Methods for multidimensional event classification: a case study using images from a Cherenkov gamma-ray telescope*. Ground-based telescopes observe high-energy gamma rays via charged particles emitted in atmospheric collisions called Cherenkov radiation. The number of collisions initiated by gamma particles observable from the ground is small compared to those initiated by hadrons (background). To effectively study this radiation, we need to discriminate the (interesting) gamma collisions from the (uninteresting) hadronic ones. The Cherenkov radiation is recorded on the telescope detector and generally takes the shape of an elongated cluster. The image is pre-processed with PCA to define an ellipse, whose properties can be used to discriminate the types of radiation caused the particle shower. These properties are the features used in our analysis. This dataset does not contain real data from the telescopes, but is generated using a Monte Carlo program simulating telescope images. This dataset also differs from real telescope data in the sense that for technical reasons, the number of hadron events is significantly underestimated.

For our analysis, we chose to compare the performance of a Naive Bayes model and an SVM. Naive Bayes is a probabilistic classification approach that uses Bayes theorem to pick the class that maximizes the posterior probability. While it makes some simplifying assumptions, it still proves to be effective for many datasets. SVMs are a more advanced classification method that finds the optimal separating hyperplane between two classes. (Citations?). We create a model using each of these methods and compare their performance on the Magic dataset, taking into account how their underlying structure affects the outcome.

2 EXPLORATORY DATA ANALYSIS

Before preprocessing the data or building our models, we performed exploratory analysis of the dataset to understand its structure and relevant statistical features. This included visualizing the data, assessing the underlying feature distributions, and determining the presence of correlated features. The results of this step determined our preprocessing strategy.

2.1 Objective

The objective of this analysis is to classify each particle shower as either caused by gamma radiation or background (hadron) radiation based on the properties of the telescope image. This is a supervised learning binary classification task.

2.2 Data Exploration

The Magic dataset has 19,020 instances, each of which has ten features and one categorical label. 12,332 of the instances are labeled gamma collisions, and 6,688 of the instances are hadronic collisions. All ten of the features are continuous, ratio-scaled, and numerical, and are related to properties of the (simulated) telescope image.

- `fLength`: major half axis of ellipse [mm]
- `fWidth`: minor half axis of ellipse [mm]
- `fSize`: 10-log of sum of content of all pixels [photon count]
- `fConc`: ratio of sum of two highest pixel over size
- `fConc1`: ratio of brightest pixel over size
- `fAsym`: distance from brightest pixel to center projected along major axis [mm]
- `fM3Long`: 3rd root of third moment along major axis [mm]
- `fM3Trans`: 3rd root of third moment along minor axis
- `fAlpha`: angle of major axis with vector to origin [deg]
- `fDist`: distance from origin to center of ellipse [mm]

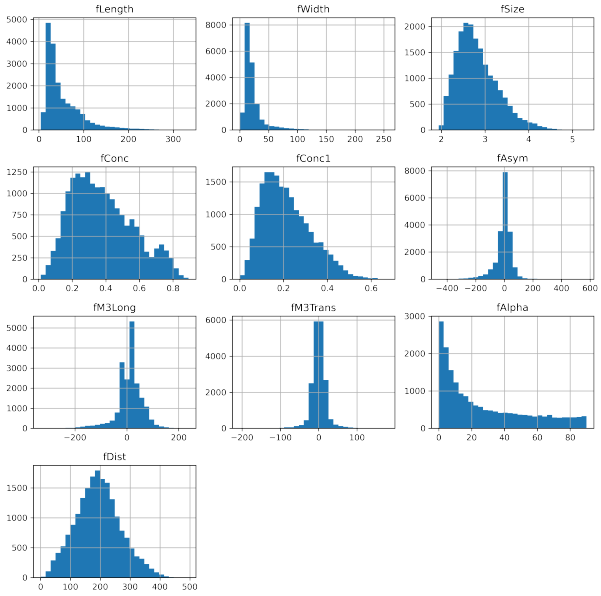

Figure 1: Distributions of all of the features. The vast majority are some form of (skewed) Gaussian distribution, with the notable exception of `fAlpha`.

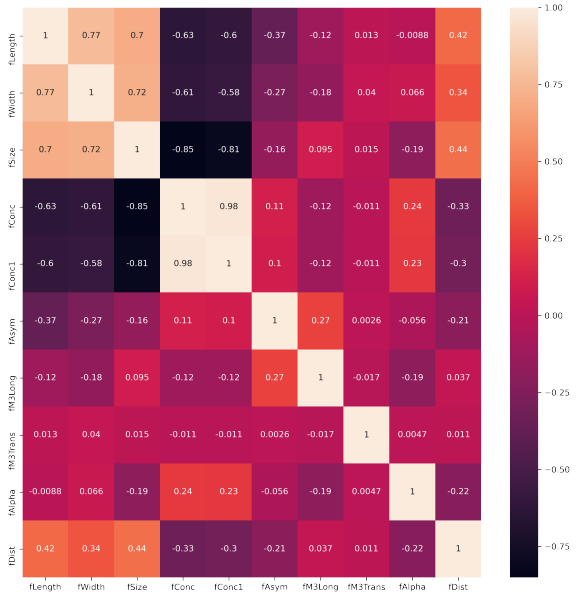

Figure 2: Correlation map between all features. Though many are correlated to a weak degree, the only strong correlations (≥ 0.8) are between `fConc`, `fConc1`, and `fSize`.

2.3 Feature Visualization

Figure 1 displays the distributions of all 10 features across the dataset. Note that most are skewed Gaussians, with the notable exceptions of `fAlpha`, and possibly `fLength` and `fWidth`. They also span varying value ranges and orders of magnitude.

Figure 2 shows a heatmap of correlations between all of the features, and figure 3 shows scatterplots of all features against each other. We can see that, while many features are weakly correlated, the only strong ($\geq$ 0.8) correlations are between `fConc`, `fConc1`, and `fSize`. Figure 3 also tells us that there are relatively few outliers compared to the number of "central" datapoints.

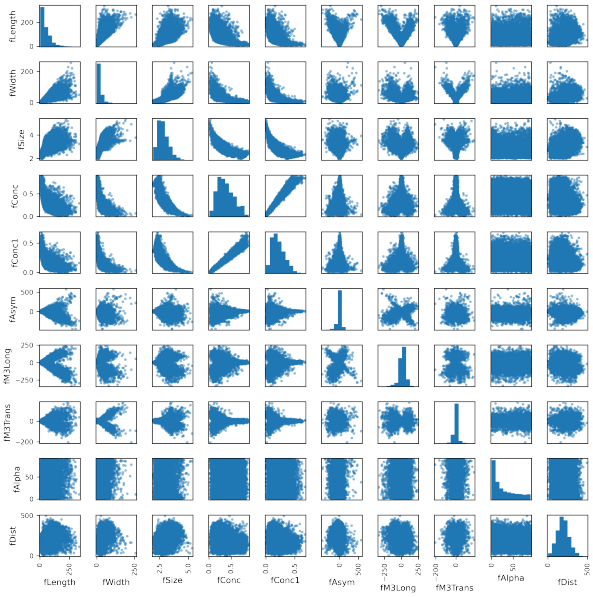

Figure 3: Scatter matrix of all features. Notice that most are correlated to some degree, but the majority only weakly. There are also very few outliers, so we don't need to worry about those.

3 DATA PREPROCESSING:

Since the dataset only contains continuous features and two labels, the only encoding necessary was binarizing the categories. There were no missing values, so data cleaning was unnecessary. Both Naive Bayes and SVM tend to work better if the features are roughly in the same range, so we performed a min-max scale to ensure that condition was met. We didn't need to remove outliers, and there aren't any clear targets for feature engineering.

We tried combinations of a number of different preprocessing steps. Naive Bayes tends to work better with uncorrelated features, so we tried removing correlations of at least 0.8 (`fConc1` and `fSize`). Since we used a Naive Bayes classifier that assumes all features are symmetric Gaussians, we tried de-skewing all of the apparently-Gaussian features with a skew of at least 0.5. The resulting six features are shown in figure 4. Surprisingly, both of these preprocessing steps decreased the accuracy of both Naive Bayes and SVM, and in the end, we discarded them.

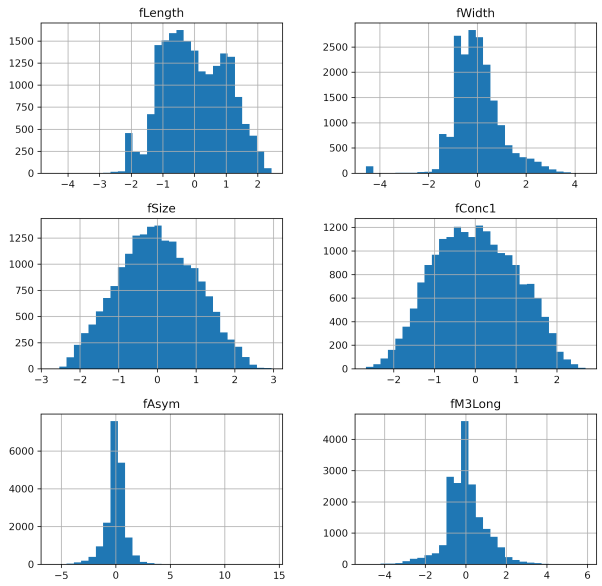

Figure 4: Six features after a de-skew process. This modification was not used in the final model, since its results were unexpectedly worse.

4 ANALYSIS AND MODELING

We chose to compare a Naive Bayes model to a SVM model. The Naive Bayes acts as our baseline model, and the SVM is our more advanced method. Both were trained and evaluated on a 75/25 (14265/4755) train-test data split.

4.1 Baseline Model: Naive Bayes

Bayes' theorem states that

$$P(c|x) = \frac{P(x|c)P(c)}{P(x)}$$

where x is a vector of features $(x_1, \ldots, x_d)$ that we want to classify, and c is a category. In general, to calculate the likelihood $P(x|c)$, we need to expand it into a product over the relevant conditional probabilities:

$$P(x|c) = \prod_{i=1}^{d} P(x_i|x_{i+1}, \ldots, x_d, c)$$

These conditional probabilities are intractable to calculate, so we make the "naive" assumption of conditional independence of features:

$$P(x|c) = \prod_{i=1}^{d} P(x_i|x_{i+1}, \ldots, x_d, c) \approx \prod_{i=1}^{d} P(x_i|c)$$

which lets us write an expression for the Naive Bayes problem in the following simpler way (dropping the normalization constant):

$$\hat{y} = \arg\max_{c} \prod_{i=1}^{d} P(x_i|c)P(c)$$

Despite this rather strong assumption, Naive Bayes has been found to work much more broadly than would "naively" be expected, as the conditional independence assumption is nearly always violated in real datasets (including Magic). Indeed, it usually works better when the assumption holds. The fact that that it's a very fast model that often works in many disparate situations makes it a good baseline to compare with the SVM.

4.2 Advanced Technique: SVMs

Support vector machines classify data by finding the hyperplane with the maximum margin between classes. This is a non-parametric approach, meaning that it makes no assumptions about the underlying data distributions. While SVMs are a linear approach, they can be used to find a nonlinear decision boundary by using the kernel trick. Even though the decision boundary is nonlinear in the input space, there may exist a linear discriminant in a higher dimensional feature space which the SVM can calculate.

SVMs are one of the more powerful classification approaches, as they can handle dependencies between features, and are effective in high dimensional feature spaces. However, they're not always appropriate, as they cannot handle datasets beyond hundreds of thousands of instances due to their quadratic computational complexity, and can perform poorly on highly skewed or imbalanced datasets (citation?). They can also be prone to overfitting if the number of parameters is large and the number of instances is small. Fortunately, the Magic dataset only has 19000 datapoints and 10 features, so neither of these drawbacks are relevant.

Choosing an appropriate kernel is essential for the success of the model. Common choices are: linear, polynomial, sigmoid, and the radial basis function (rbf), but custom kernels are sometimes used instead. The rbf kernel proved to be the most effective of the ones we tested. Other parameters include C, which trades off misclassification rate against simplicity of decision boundary; and γ, which determines the influence of a single instance. Experimentation led to choosing $C = 60$ and $\gamma = 1/(n_{\text{features}} * \sigma^2(X))$ for our model.

5 MODEL EVALUATION

5.1 Evaluation Metrics

In binary classification, one category is generally labeled as the positive category and the other category is labeled as the negative. Most binary classification methods generate a real number, usually interpreted as a probability, but which regardless tracks the likelihood of an instance belonging to the positive class. Depending on the task, it may be important to limit false negatives, or it may be more important to limit false positives. In our case, it is more important not to misclassify background as signal than it is to correctly classify signal. Therefore, the accuracy score alone is not a good indicator of the success of the model, since one may have a high accuracy with a high number of false positives.

For our analysis, we generated four metrics to compare our models. Of most significance are the area under the ROC curve and the area under the precision-recall curve. We also calculated the accuracy and F1 score in order to compare the performance of our models to that of other work on this dataset.

5.1.1 Accuracy. The accuracy is the fraction of correctly classified samples, with 1 being perfect classification.

$$\text{Accuracy} = \frac{TP + TN}{n}$$

where TP stands for the number of true positives, TN stands for the number of true negatives, and n is the total number of samples.

5.1.2 F-Score. The F-score is a weighted average of precision and recall. A perfect classifier has an F-score of 1.

$$F = \frac{1}{2} \sum_{i=1}^{2} \frac{2 * \text{prec}_i \cdot \text{recall}_i}{\text{prec}_i + \text{recall}_i}$$

5.1.3 Area under ROC Curve. The area under the ROC curve is one of the best methods of comparing classification models, since it is independent of criterion (the percentage of acceptable false positives). The ROC curve is the plot of the false positive rate against the true positive rate for decreasing criterion, and the area under the ROC curve indicates classifier performance, with 0.5 being random classification and 1 being perfect classification.

5.1.4 Area under Precision-Recall Curve. The area under the ROC curve is a sufficient evaluation metric for most datasets, but if the dataset has significantly more negative examples than positive examples, then the precision-recall curve is a better measure of performance. While not necessary for the Magic dataset, we chose to calculate it for comparison. The P-R Curve plots the precision for the positive class against the recall for the positive class.

$$\text{precision} = \frac{TP}{TP + FP}$$

$$\text{recall} = \frac{TP}{TP + FN}$$

The area under the precision recall curve is an indicator of classifier performance with 1 being perfect classification.

5.2 Results

As shown in table 1, the SVM model clearly outperformed Naive Bayes in all metrics. It particularly did better in the F-score and area under the precision accuracy curve, indicating that the Naive Bayes model may have been more susceptible to false positives. Figure 5 shows the ROC and Precision-Recall curves for each model, which display visualy the large difference in model quality. SVM produces a smooth curve that approaches the upper corner of the plot (left for ROC and right for PR) which is what you would expect of a good model. In contrast, Naive Bayes isn't particularly smooth and is closer to the diagonal.

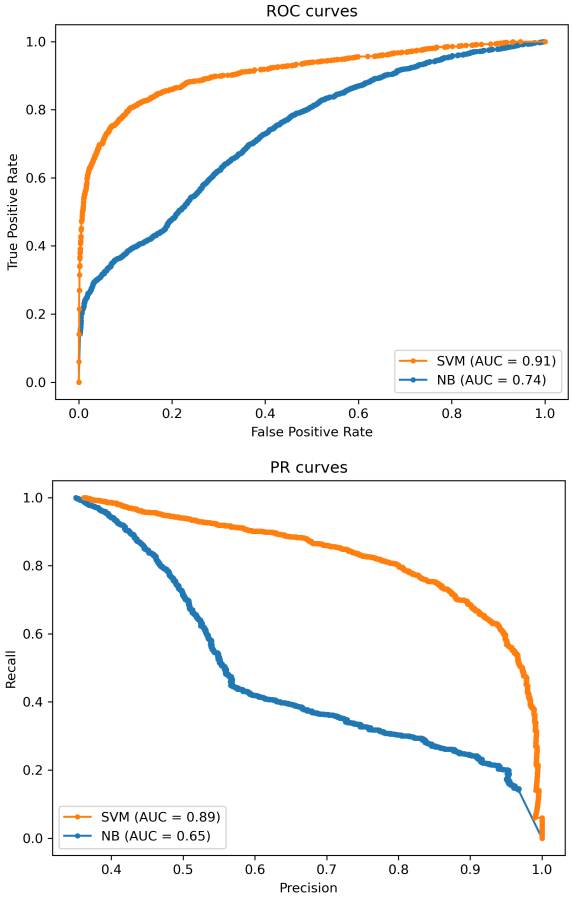

Figure 5: The ROC and Precision-Recall curves for both models. SVM outperforms Naive Bayes on both metrics (the area under the curves).

	Naive Bayes	SVM
Accuracy	0.73	0.86
F-score	0.49	0.78
AUC ROC	0.78	0.92
AUC PR	0.71	0.91

Table 1: Model evaluation results for Naive Bayes and SVM using accuracy, f-score, area under the ROC curve and area under the precision-accuracy curve. SVM outperforms Naive Bayes on all metrics.

6 DISCUSSION

For the Magic Dataset, the SVM appears to be the superior to Naive Bayes for binary classification. This is likely due to the properties of the dataset and the approaches (write this better). The Magic dataset has many features that show some sort of dependency and two features which appear to be directly correlated. It has far more instances than features, but not hundreds of thousands of instances. Its features are primarily normally distributed, and the number of positive and negative samples is not particularly unbalanced. All of these are reasons why SVM would be a good choice of model. SVMs can handle feature dependencies, and work well in high dimensions. There are ratio of instances to features is high enough that overfitting is unlikely, and there are few enough instances for the model to train in a reasonable amount of time. Naive Bayes is not necessarily a bad choice of model for this dataset, but it does make the assumption that features are independent. We believe that this assumption may be responsible for the poorer performance. If

the independence assumption held, you could likely expect much better performance.

While SVMs were the better choice with this dataset, that isn't always the case. SVMs might not be a good choice if there were very few instances or very many instances as this could cause overfitting or intractable training time respectively. In the event that the independence assumption holds and there are hundreds of thousands of training samples, Naive Bayes would be the preferable classifier as it can operate much more efficiently.

One point of confusion we faced in this analysis was that more advanced pre-processing methods seemed to decrease performance. We attempted normalizing the variables to the standard normal distribution and adjusting their skew, but these attempts decreased performance. The best performance was found with standardizing all features to be between 0 and 1. We are still not entirely sure why this would be, but it may just show that sometimes manipulating the data too much may interfere with the models ability to work with the data.

7 CONCLUSIONS

In this analysis, we compared the binary classification performance of Naive Bayes and SVMs on the Magic Dataset, finding that the SVM model performed much better than Naive Bayes in every metric. We believe that this is largely due to the fact that the independence assumption for Naive Bayes did not hold. Future work could involve improving the preprocessing or constructing a kernel specific to this task for the SVM, instead of using the radial basis function kernel. We could also compare other approaches, such as neural networks or decision trees.

ACKNOWLEDGMENTS

We would like to thank Professor Thaddius Francis Pawlicki for his informative lectures and teaching during this strange semester, and the CSC 240 TA's for their consistent help and guidance.

REFERENCES

[1] Mohammed J. Zaki and Jr. Wagner Meira. 2020. *Data Mining and Machine Learning: Fundamental Concepts and Algorithms* (2. ed.). Cambridge University Press, USA.