Capstone Project: Predicting the class of events from a simulated MAGIC gamma telescope

by Shaun Lippy (Udacian)

I. Summary of Project and Reasoning Behind Choice of Data set

This data set was acquired from the University of California at Irvine Machine Learning Repository[1], which is an extensive collection of data sets provided for general use in order to facilitate education in ML. The data set is a simulated collection of events that mimic what would be observed through a MAGIC gamma telescope. The simulation was done using Monte Carlo methods – more details can be obtained from the web site.

The problem is one of supervised learning and, more specifically, one of classification. The goal is to distinguish between a true signal (referred to as a "gamma" event) and background noise (a "hadron" event). Machine learning is an ideal tool for this type of problem because it can help scientists eliminate much of the noise in a data set and allow them to focus on a much smaller subset of the total number of events, thereby using their time more efficiently and effectively.

I have chosen this data set because one of my primary interests in studying machine learning is to understand how it can be used to help advance the frontier of scientific knowledge, something which is very important and fascinating to me. I am particularly interested in astronomy and space science, and a peripheral goal is to advance my skills in the field of data science in order to one day pursue a career change that will allow me to work in the field of scientific research. With that goal in mind, this project is both exciting and valuable to me!

II. Statistical Analysis and Data Exploration

Number of data points?

The number of data points in the data set is 19,020.

Number of features?

The number of features in the data set is 10.

Feature statistics

Table 1 provides an overview of the basic statistical properties of the data set. It is notable that the scale of several of the features (fSize, fConc, and fConc1) is quite a bit smaller than that of the other features, and as such it seems that it will be prudent to normalize (center) the data so that all columns have a mean of 0 and a variance of 1. This will be done in the preprocessing step.

[1] https://archive.ics.uci.edu/ml/datasets/MAGIC+Gamma+Telescope

FEATURE	MIN	MAX	MEAN	MEDIAN	STD
fLength	4.28	334.18	53.25	37.15	42.36
fWidth	0.00	256.38	22.18	17.14	18.35
fSize	1.94	5.32	2.83	2.74	0.47
fConc	0.01	0.89	0.38	0.35	0.18
fConc1	0.00	0.68	0.21	0.20	0.11
fAsym	-457.92	575.24	-4.33	4.01	59.21
fM3Long	-331.78	238.32	10.55	15.31	51.00
fM3Trans	-205.89	179.85	0.25	0.67	20.83
fAlpha	0.00	90.00	27.65	17.68	26.10
fDist	1.28	495.56	193.82	191.85	74.73

Table 1 – Basic statistical analysis of the data set

Principal Component Analysis was run, using RandomizedPCA in sklearn, in order to get an idea of what kind of dimensionality reduction might be accomplished, if needed. The results of this analysis can be seen in Figure 1. It appears that there are 3 principal components arising from the analysis, with the top 3 components accounting for roughly 90 percent of the explained variance ratio.

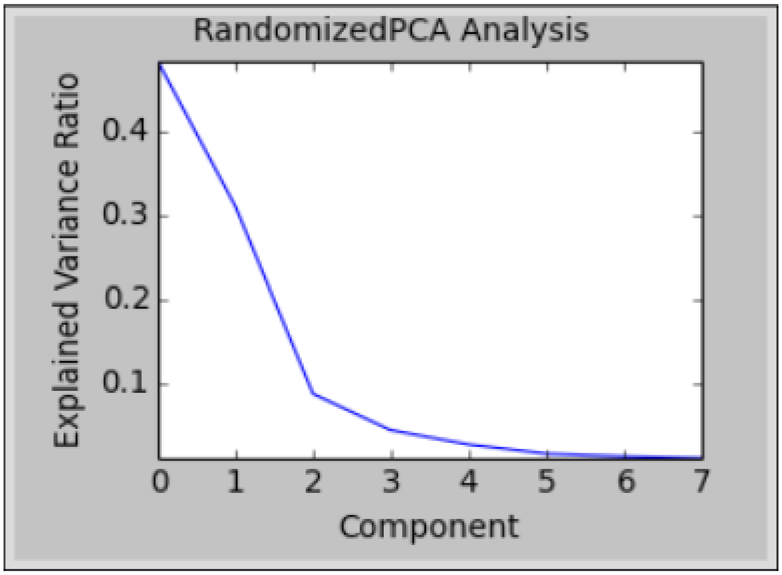

Figure 1 – Principal Component Analysis of the dataset

III. Evaluating Model Performance

It is explicitly called out in the notes to the data set that “simple classification accuracy is not meaningful for this data, since classifying a background event as signal is worse than classifying a signal event as background."[2] With this in mind, each classifier will individually be cross-validated using *roc_auc_score* as the scoring metric. This metric uses the area under the ROC (Receiver Operating Characteristic) curve, which plots the the False Positive Rate (on the abscissa) vs. the True Positive Rate (on the ordinate). A larger area under this curve indicates a superior classifier. Once each classifier has been trained and cross-validated (using K-Fold and grid search), a graph of the respective ROC curves will be plotted so that a visual analysis of the classifiers can be made and the best one selected.

IV. Analyzing Model Performance

From the plot of ROC curves (see Figure 2) we can see that several of the classifiers performed relatively well (with greater than 0.8 for the *roc_auc_score*). The Random Forest Classifier and the Support Vector Machine had nearly identical scores, and both performed slightly better than the others. The Logistic Regression classifier performed poorest and would not be considered a good solution even when taking into account its small training times (see below). The scoring results are shown below in Table 2 for a given run (due to randomness in the creation of the training and test sets, and in the training of the classifier, the results vary from run to run; but the relative performance remains largely consistent).

CLASSIFIER	PARAMETERS	roc_auc_score
Random Forest	criterion=entropy, max_depth=9, n_estimators=9	0.827172
Logistic Regression	C=100.0, solver=liblinear	0.748448
Support Vector Machine	degree=2, kernel=rbf, C=1000.0	0.824769
K-Nearest Neighbors	algorithm=ball_tree, leaf_size=10, n_neighbors=10, weights=uniform	0.803162

Table 2 – Classifier description and roc_auc_score scores

Another factor to consider, when selecting a classifier, is the time to train the classifier, and the time it takes to make predictions. The training-testing split was done so that 80% of the data points were used for training and cross-validation, while 20% of the samples were used for testing. Table 3 shows the resulting training and testing times for the four classifiers explored.

CLASSIFIER	TRAINING TIME	TESTING TIME
Random Forest	50.638624	0.004718
Logistic Regression	2.919238	0.001415
Support Vector Machine	380.375287	0.746997
K-Nearest Neighbors	61.473052	0.317583

Table 3 – Training and testing times for each classifier

[2] https://archive.ics.uci.edu/ml/machine-learning-databases/magic/magic04.names

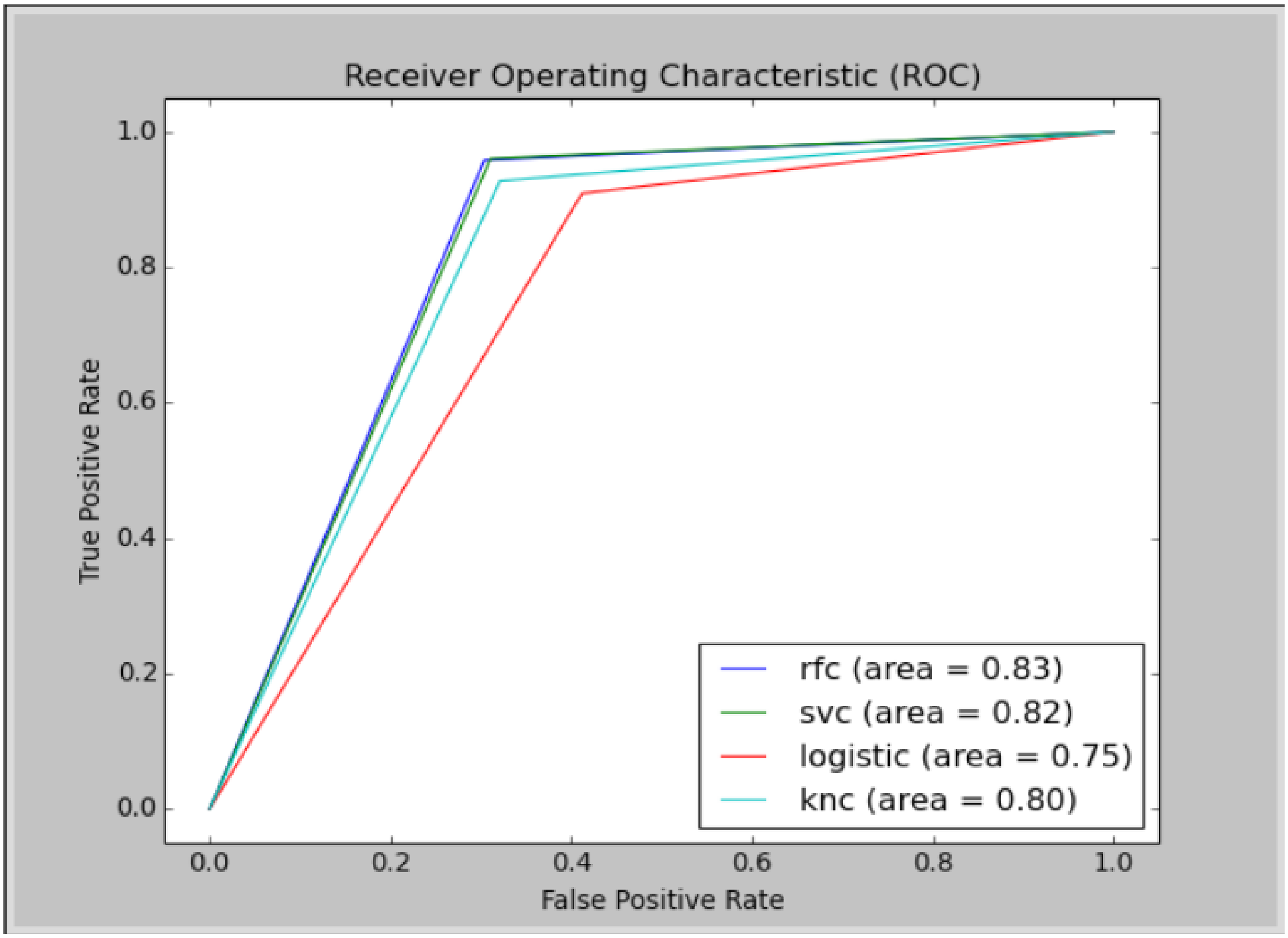

Figure 2 – ROC curve comparing the different classifiers

Clearly, the Support Vector Machine takes much longer than the other classifiers to train, while the Logistic Regression model is by far the quickest (but also the least accurate in terms of roc_auc_score). The K-Nearest Neighbors and Random Forest models take similar amounts of time to train. In terms of testing time, Random Forest and Logistic Regression are both noticeably faster than the other two classifiers, and Support Vector Machine is about 2.5x slower than K-Nearest Neighbors.

V. Conclusions

When viewing the classifiers by way of comparison using all of the above statistics and results, it is clear that the best model for this problem is the **Random Forest Classifier**. The Random Forest is the most accurate in terms of *roc_auc_score*, although the Support Vector Machine comes close in terms of the scoring. However, the Random Forest outperforms the Support Vector Machine significantly when it comes to training and testing times, as can be seen in Table 3. This classifier is, I believe, successful and could be recommended to the scientists looking to classify data from a MAGIC gamma telescope!