CASE STUDY: CLASSIFICATION OF WINNING AND LOSING FUNDS

The following case study was developed and written by Matthew Dixon, PhD, FRM.

A research analyst for a fund of funds has been tasked with identifying a set of attractive exchange-traded funds (ETFs) and mutual funds (MFs) in which to invest. She decides to use machine learning to identify the best (i.e., winners) and worst (i.e., losers) performing funds and the features which are most important in such an identification. Her aim is to train a model to correctly classify the winners and losers and then to use it to predict future outperformers. She is unsure of which type of machine learning classification model (i.e., classifier) would work best, so she reports and cross-compares her findings using several different well-known machine learning algorithms.

The goal of this case is to demonstrate the application of machine learning classification to fund selection. Therefore, the analyst will use the following classifiers to identify the best and worst performing funds:

- classification and regression tree (CART),

- support vector machine (SVM),
- *k*-nearest neighbors (KNN), and
- random forests.

Data Description

In the following experiments, the performance of each fund is learned by the machine learning algorithms based on fund type and size, asset class composition, fundamentals (i.e., valuation multiples), and sector composition characteristics. To form a cross-sectional classifier, the sector composition and fund size reported on 15 February 2019 are assumed to be representative of the latest month over which the fund return is reported. Exhibit 12 presents a description of the dataset.

Exhibit 12 Dataset Description

Dataset: MF and ETF Data

There are two separate datasets, one for MFs and one for ETFs, consisting of fund type, size, asset class composition, fundamental financial ratios, sector weights, and monthly total return labeled to indicate the fund as being a winner, a loser, or neither. Number of observations: 6,085 MFs and 1,594 ETFs.

Features: Up to 21, as shown below:

General (six features):

1. cat_investment*: Fund type, either "blend," "growth," or "value"
2. net_assets: Total net assets in US dollars
3. cat_size: Investment category size, either "small," "medium," or "large" market capitalization stocks
4. portfolio_cash**: The ratio of cash to total assets in the fund
5. portfolio_stocks: The ratio of stocks to total assets in the fund
6. portfolio_bonds: The ratio of bonds to total assets in the fund

Fundamentals (four features):

1. price_earnings: The ratio of price per share to earnings per share
2. price_book: The ratio of price per share to book value per share
3. price_sales: The ratio of price per share to sales per share
4. price_cashflow: The ratio of price per share to cash flow per share

Sector weights (for 11 sectors) provided as percentages:

1. basic_materials
2. consumer_cyclical
3. financial_services
4. real_estate
5. consumer_defensive
6. healthcare
7. utilities
8. communication_services
9. energy
10. industrials
11. technology

Exhibit 12 (Continued)

Labels:

Winning and losing ETFs or MFs are determined based on whether their returns are one standard deviation or more above or below the distribution of one-month fund returns across all ETFs or across all MFs, respectively. More precisely, the labels are:

1, if fund_return_1 month ≥ mean(fund_return_1 month) + one std.dev(-fund_return_1 month), indicating a winning fund;

-1, if fund_return_1 month ≤ mean(fund_return_1 month) – one std.dev(-fund_return_1 month), indicating a losing fund; and

0, otherwise.

*Feature appears in the ETF dataset only.
**Feature appears in the MF dataset only.
Data sources: Kaggle, Yahoo Finance on 15 February 2019.

Methodology

The classification model is trained to determine whether a fund's performance is one standard deviation or more above the mean return (Label 1), within one standard deviation of the mean return (Label 0), or one standard deviation or more below the mean return (Label -1), where the mean return and standard deviation are either for all ETFs or all MFs, depending on the particular fund's type (ETF or MF). Performance is based on the one-month return of each fund as of 15 February 2019.

This procedure results in most of the funds being labeled as "0" (or average). After removing missing values in the dataset, there are 1,594 and 6,085 observations in the ETF and MF datasets, respectively. The data table is a 7,679 × 22 matrix, with 7,679 rows for each fund observation (1,594 for ETFs and 6,085 for MFs) and 22 columns for the 21 features plus the return label, and all data are recorded as of 15 February 2019.

The aim of the experiment is to identify not only winning and losing funds but also the features which are useful for distinguishing winners from losers. An important caveat, however, is that no claim is made that such features are causal.

A separate multi-classifier, with three classes, is run for each dataset. Four types of machine learning algorithms are used to build each classifier: (i) CART, (ii) SVM, (iii) KNN, and (iv) random forest. Random forest is an example of an ensemble method (based on bagging), whereas the other three algorithms do not use bagging.

A typical experimental design would involve using 70% of the data for training and holding 15% for tuning model hyperparameters and the remaining 15% of the data for testing. For simplicity, we shall not tune the hyperparameters but simply use the default settings without attempting to fine tune each one for best performance. So, in this case, we do not withhold 15% of the data for validation but instead train the classifier on a random split of 70% of the dataset, with the remaining 30% of the dataset used for testing. Crucially, for fairness of evaluation, each algorithm is trained and tested on identical data: The same 70% of observations are used for training each algorithm, and the same 30% are used for testing each one. The most important hyperparameters and settings for the algorithms are shown in Exhibit 13.

Exhibit 13 Parameter Settings for the Four Machine Learning Classifiers

1 CART: maximum tree depth: 5 levels

(continued)

Exhibit 13 (Continued)

2 SVM: cost parameter: 1.0

3 KNN: number of nearest neighbors: 4

4 Random forest: number of trees: 100; maximum tree depth: 20 levels

The choices of hyperparameter values for the four machine learning classifiers are supported by theory, academic research, practice, and experimentation to yield a satisfactory bias–variance trade-off. For SVM, the cost parameter is a penalty on the margin of the decision boundary. A large cost parameter forces the SVM to use a thin margin, whereas a smaller cost parameter widens the margin. For random forests, recall that this is an ensemble method which uses multiple decision trees to classify, typically by majority vote. Importantly, no claim is made that these choices of hyperparameters are universally optimal for any dataset.

Results

The results of each classifier are evaluated separately on the test portion of the ETF and MF datasets. The evaluation metrics used are based on Type I and Type II classification errors, where a Type I error is a false positive (FP) and a Type II error is a false negative (FN). Correct classifications are true positive (TP) and true negative (TN).

- The first evaluation metric is **accuracy**, the percentage of correctly predicted classes out of total predictions. So, high accuracy implies low Type I and Type II errors.
- **F1 score**, the second evaluation metric, is the weighted average of precision and recall. Precision is the ratio of correctly predicted positive classes to all predicted positive classes, and recall is the ratio of correctly predicted positive classes to all actual positive classes.

F1 score is a more appropriate evaluation metric to use than accuracy when there is unequal class distribution ("class imbalance") in the dataset, as is the case here. As mentioned, most of the funds in the ETF and MF datasets are designated as "0," indicating average performers.

Exhibit 14 shows the comparative performance results for each algorithm applied to the ETF dataset. These results show the random forest model is the most accurate (0.812), but once class imbalance is accounted for using F1 score (0.770), random forest is about as good as CART. Generally, ensemble methods, such as random forest, are expected to be at least as good as their single-model counterparts because ensemble forecasts generalize better out-of-sample. Importantly, while the relative accuracies and F1 scores across the different methods provide a basis for comparison, they do not speak to the absolute performance. In this regard, values approaching 1 suggest an excellent model, whereas values of approximately 1/3 would indicate the model is useless: 1/3 is premised on three (+1, 0, -1) equally distributed labels. However, because the distribution of classes is often not balanced, this ratio typically requires some adjustment.

Exhibit 14 Comparison of Accuracy and F1 Score for Each Classifier Applied to the ETF Dataset

	CART	SVM	KNN	Random Forest
Accuracy	0.770	0.774	0.724	0.812
F1 score	0.769	0.693	0.683	0.770

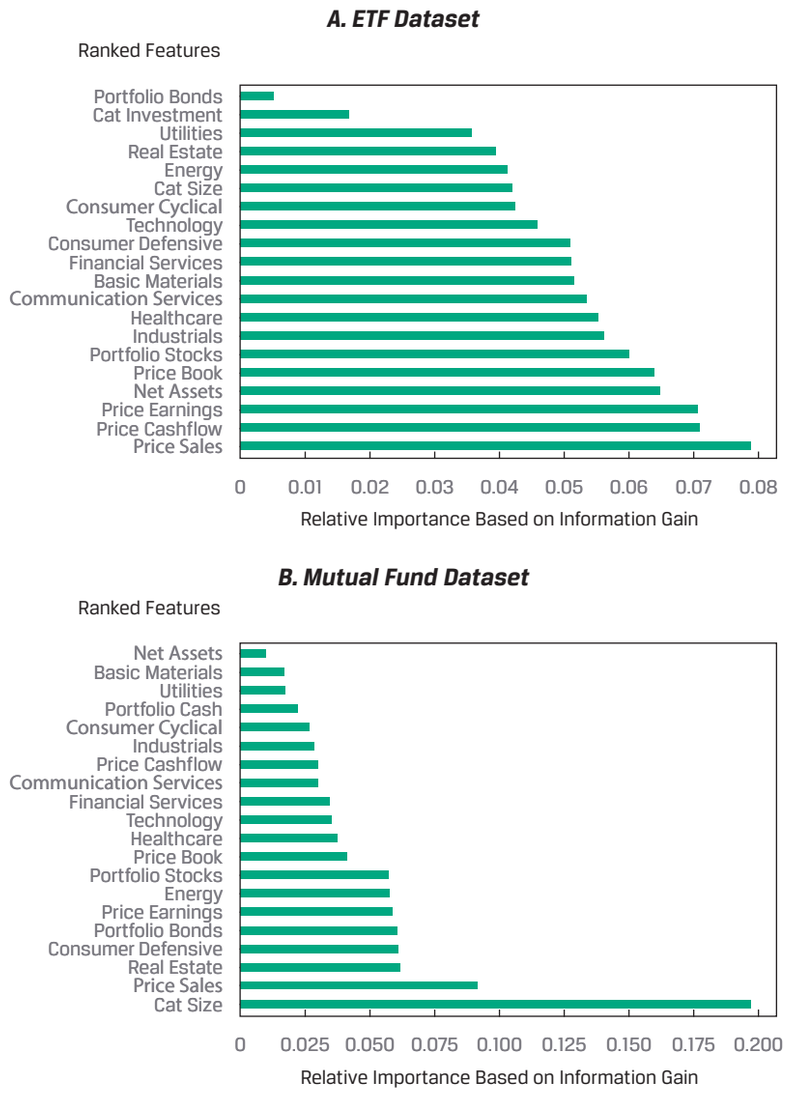
Exhibit 15 shows that the random forest model outperforms all the other classifiers under both metrics when applied to the MF dataset. Overall, the accuracy and F1 score for the SVM and KNN methods are similar for each dataset, and these algorithms are dominated by CART and random forest, especially in the larger MF dataset. The difference in performance between the two datasets for all the algorithms is to be expected, since the MF dataset is approximately four times larger than the ETF dataset and a larger sample set generally leads to better model performance. Moreover, the precise explanation of why random forest and CART outperform SVM and KNN is beyond the scope of this case. Suffice it to say that random forests are well known to be more robust to noise than most other classifiers.

Exhibit 15 Comparison of Accuracy and F1 Score for Each Classifier Applied to the Mutual Fund Dataset

	CART	SVM	KNN	Random Forest
Accuracy	0.959	0.859	0.856	0.969
F1 score	0.959	0.847	0.855	0.969

Exhibit 16 presents results on the relative importance of the features in the random forest model for both the ETF (Panel A) and MF (Panel B) datasets. Relative importance is determined by **information gain**, which quantifies the amount of information that the feature holds about the response. Information gain can be regarded as a form of non-linear correlation between Y and X. Note the horizontal scale of Panel B (MF dataset) is more than twice as large as that of Panel A (ETF dataset), and the bar colors represent the feature rankings, not the features themselves.

Exhibit 16 Relative Importance of Features in the Random Forest Model



The prices-to-sales (price_sales) and prices-to-earnings (price_earnings) ratios are observed to be important indicators of performance, at about 0.08–0.09 and 0.06–0.07, respectively, in the random forest models for each dataset. The ratio of stocks to total assets (portfolio_stocks), at 0.06, is another key feature. Moreover, the industrials, health care, and communication services sector weightings are relatively important in the ETF dataset, while the real estate, consumer defensive, and energy sector weightings are key features in the MF dataset for differentiating between winning and losing funds.

Another important observation is that the category of the fund size (cat_size) is by far the most important feature in the model's performance for the MF dataset (≈ 0.20), whereas it is of much less importance for model performance using the ETF dataset (≈ 0.04). Conversely, net assets is a relatively important feature for model performance using the ETF dataset (0.065), while it is the least important feature when the random forest model is applied to the MF dataset (0.01).

Conclusion

The research analyst has trained and tested machine learning–based models that she can use to identify potential winning and losing ETFs and MFs. Her classification models use input features based on fund type and size, asset class composition, fundamentals, and sector composition characteristics. She is more confident in her assessment of MFs than of ETFs, owing to the substantially larger sample size of the former. She is also confident that any imbalance in class has not led to misinterpretation of her models' results, since she uses F1 score as her primary model evaluation metric. Moreover, she determines that the best performing model using both datasets is an ensemble-type random forest model. Finally, she concludes that while fundamental ratios, asset class ratios, and sector composition are important features for both models, net assets and category size also figure prominently in discriminating between winning and losing ETFs and MFs.