Feature Selection in Finance

Benjamin A. Schifman
Justin J. Siekmann
Department of Electrical and Computer Engineering
University of Arizona
Tucson, AZ 85719
bschifman@email.arizona.edu
jsiekmannemail.arizona.edu

Abstract

There exist technical analysis indicators traditionally used by analysts to evaluate and predict market and equity performance. These indicators provide a unique perspective on the strength and direction of the underlying price action in market data. Being able to identify the most relevant technical analysis indicators through the use of feature selecting methods offers for a more optimized strategy in quantitatively computing financial estimation models.

1 Introduction

Feature selection is used to determine relevant indicators while identifying those that are irrelevant and redundant. This paper analyzes multiple implementations and approaches of feature selection and dimensionality reduction in the form of: Feature Correlation, Maximal Information Coefficient, Recursive Feature Elimination, and Principal Component Analysis. Optimizing algorithms based on these reduced indicator sets could be used to predict performance of individual equities, sectors, or overall markets along with classifying and identifying the correlation and interdependencies between them. The goal of this paper is to implement various methods of feature selection to determine the efficacy of specific technical indicators as enablers to successful financial analysis. From this project we hope to deepen our understanding of the usage cases for applying specific machine learning algorithms as well as expanding upon our technical analysis of the stock market and which indicators play a role in successful market analysis.

2 Methods/Approach

The following subsections present details and explanations of the methods and functions implemented as part of this project.

2.1 Data and Technical Analysis Indicators

The Quandl platform was used to fetch 11 years of market data in total from Dec 31, 2006 to March 27, 2018 on various identified US tickers across different sectors. Tickers used in the project can be found in Table 1 categorized by sector. As the process to fetch and preprocess the data is time consuming, pickle files were used to save data locally to be quickly reimported. TA-Lib: Technical Analysis Library was used to calculate features on the market data for each ticker. TA-Lib has the ability to calculate many technical analysis indicators in various categories. The features incorporated into this project, found with in Table 2, are a selected subset of the indicators offered in TA-lib based upon the categories in TA-Lib, popularity online, and expert's favorites and essentials.

ECE523: Engineering Applications of Machine Learning and Data Analytics (Sp2018).

Financials	Utilities	Energy	Healthcare	Technology	Real Estate
JPM	T	XOM	JNJ	AAPL	ECL
BAC	VZ	CVX	UNH	GOOGL	FMC
WFC	NEE	BP	PFE	MSFT	IP
C	TMUS	GE	MRK	FB	VMC
MS		SLB	ABBV	INTC	BMS
			MMM	CSCO	
			AMGN	ORCL	
			MDT	IBM	
				NVDA	

Table 1: Tickers

Overlap	Momentum	Volume	Cycle	Price	Volatility	Statistical	Math Transformation	Pattern Recognition
SMA	RSI	OBV	HT_DCPERIOD	AVGPRICE	ATR	BETA	EXP	CDLENGULFING
EMA	MOM	AD	HT_TRENDMODE	TYPPRICE		LINEARREG	LN	CDLDOJI
BBAND_Upper	ROC	ADOSC				VAR		CDLHAMMER
BBAND_Middle	ROCP							CDLHANGINGMAN
BBAND_Lower	STOCH_SLOWK							
HT_TRENDLINE	STOCH_SLOWD							
SAR	MACD							
	MACDSIGNAL							
	MACDHIST							

Table 2: Technical Analysis Indicators

2.1.1 Indicator Categories

Overlap indicators generally can be overlayed onto the price charts. They most commonly include different styles of moving average calculations.
Momentum indicators convey how quickly the price of the ticker is moving. For example, the faster the price of a ticker increases the larger its momentum.
Volume indicators take into account the volume of the day's trading into account.
Cycle indicators attempt to identify changes in the overall direction of the ticker's movement.
Price indicators combine the multiple prices in the data into one value.
Volatility indicators convey how sporadic the ticker's prices are.
Statistical indicators are based on statistical concepts and can be used in a number of different ways.
Math Transformation indicators apply common mathematical operations upon the tickers prices.
Pattern Recognition indicators look for patterns in the prices of a ticker traders have identified are indicative of future outcomes.

2.2 Normalization

As each technical analysis indicator produces values applicable based on how the indicator was calculated, normalization of the indicators makes correlations between them during feature selection more accurate and applicable. Each value is normalized using Equation (1).

$$x_n = \frac{x - min}{max - min} \tag{1}$$

2.3 Feature Correlation

Removing highly correlated features allows for the optimization of the classification algorithms by reducing the feature space. Features that are highly correlated most likely offer no additional data and they are an extra expense in computation time. The pairwise correlation of columns was computed and heat mapped in Figure 1.

2.4 Maximal Information Coefficient (MIC)

The MIC is a measure of two-variable dependence designed specifically for rapid exploration of many-dimensional data sets [1]. A benefit to MIC correlations between two variables is that it can be described regardless of linear or non-linear relationships. The MIC yields a single vale $0 \leq MIC \leq 1$ with a value closer to 1 representing that the variables are more closely correlated, and a value near

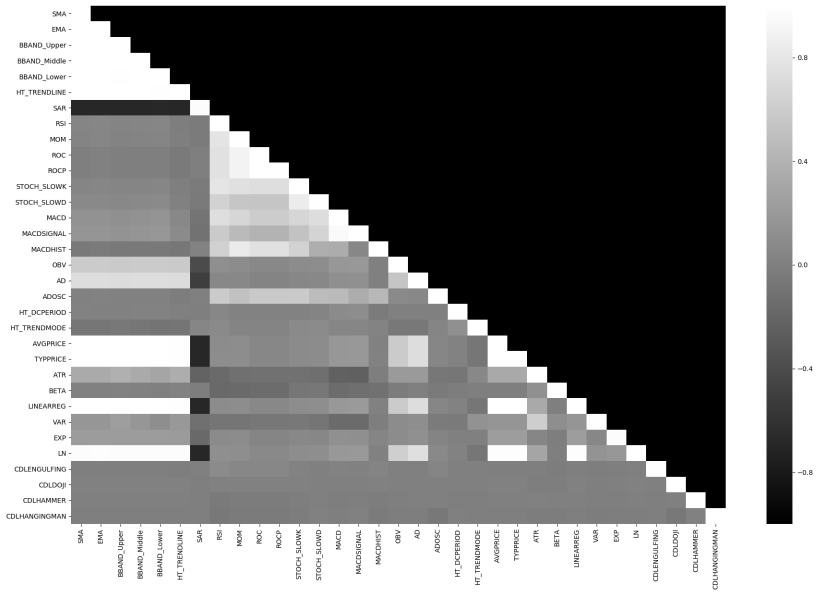

Figure 1: Heat Map of Correlated Variables

0 indicates statistically independent variables that have neither linear nor nonlinear relationships. The $minepy$ library was used in python to rank the features according to their MIC with the target variable. The MIC was calculated for each feature in each ticker, yielding 33 MIC values for each ticker, and then a final set of MIC values were calculated by taking the ticker-wise mean over the MIC values.

2.5 Recursive Feature Elimination (RFE)

RFE is a method of recursively selecting smaller subsets of the larger feature set to then be used in an external classifier. The goal of RFE is to reduce the feature set into the smallest set of relevant and valuable features which yield the greatest accuracy. The time to run of the external classifier is reduced due to the fact that there are fewer features analyzed in the classification. Given a base classifier, recursive feature elimination can be performed by re-training copies of the base classifier using a certain number of dropped features per iteration until the specified number of features to keep has been hit. RFE was implemented independently with two different base classifiers, SVM and Adaboosted Decision Trees.

2.6 Random Forest Classifier (RFC)

The RFC is an ensemble algorithm implementing a combination of decision trees classifiers where the majority vote of all trees are used to classify the input feature vector [2]. This RFC bagging method was used in order classify the input data, while also intrinsically implementing feature selection to return a vector of feature importances. The RFC was chosen due to its implementation of bagging, and the entire feature set was fed into the RFC while using 10 decision trees. The main focus of using the RFC was to determine the vector of feature importance, while the accuracy of the algorithm was secondary.

2.7 Principal Component Analysis (PCA)

PCA orthogonally transforms a set of features into a set of linearly uncorrelated principal components. PCA is a method for reducing the dimensionality of the feature set size while retaining principal

component variance, and the features informational relevance. PCA was implemented on the original 33 features and then the distilled principal components were used in the RFC classifier with 3 fold cross validation to determine the accuracy. The RFC classifier was used to analyze the validity of PCA because it intrinsically implements a form of feature selection as it weighs the features/principal components in terms of importance. The number of principal components the feature set was to be reduced to was iterated from 1 to 33 and then the accuracy of the RFC algorithm was calculated on each reduced feature set/principal component set.

2.8 Multi Layer Perceptron (MLP)

Training a multi layer perceptron inherently weights the input features and allows for complex interdependencies to be computed within the network. Three models were built using keras with three layers, 50 hidden nodes using the hyperbolic tangent activation function, and an output layer using the softmax activation function. Each model is trained on different feature sets. The first is trained using all 33 features, however the second and third only consist of five. The second MLP takes handpicked features selected as those with the highest confidence by experts. The third MLP uses the five features selected by the RFE-Adaboosted Decision Trees selector.

3 Results

Two testing sets were used to obtain results in this project. First the data was split into two sections. The first set consists of the majority of data (Dec 31, 2006 to Dec 31, 2016). This first set was randomly split using sklearn's train_test_split function to train and test each approach. Therefore the test set split from the first set is a random collection of 30% of the days with the associated labels. The second set (Jan 1, 2017 to Mar 27, 2018) was only used in testing and tests performance day to day over a large chunk of time untouched by any training. As our random forest approach used cross fold validation to test and retrain, this second data set was not used as a secondary test for our random forest or PCA approaches.

3.1 Maximal Information Coefficient (MIC)

The results from the MIC algorithm are depicted in the following Figure 2. From observing this outputted MIC data it would appear that the following technical indicators: SAR, HT_TREND, CDLENGULFING, CDLDOJI, CDLHAMMER, and CDLHANGINGMAN carry minimal importance in regards to the MIC, where as all of the other technical indicators have a relatively similar resulting MIC value. This conclusion, however, holds very little relevance when trying to implement in the various types of feature selection as it provides insufficient granularity of detail to properly select from the remaining feature set. Also, the removed MIC values hold value in the other feature selection techniques, and it would seem that the calculation of the MIC is biased against pattern recognition technical indicators as it suggested that these pattern recognition techniques were negligible.

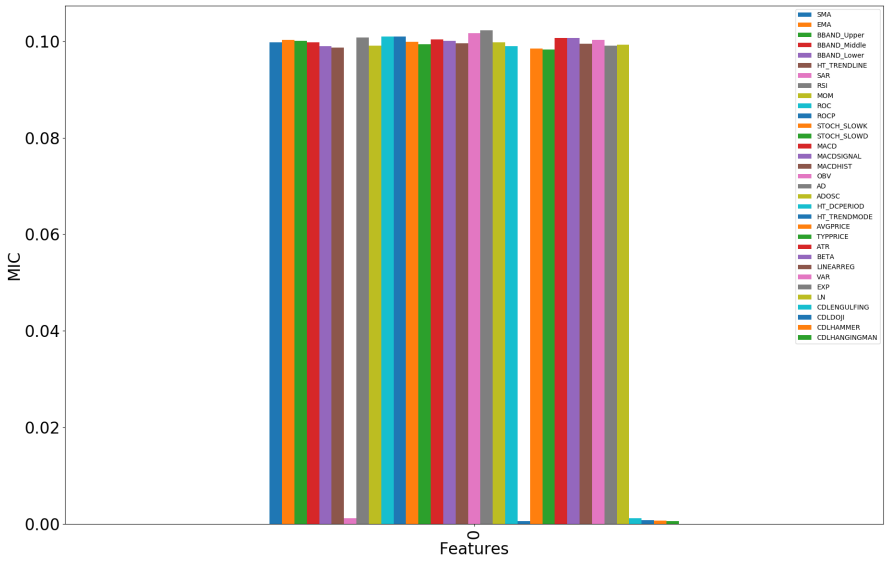

Figure 2: Maximal Information Coefficient

3.2 Recursive Feature Elimination (RFE)

The RFE approaches were run multiple times to select 5, 10, and 15 final features. The results of both RFE-SVM and RFE-Adaboost were then plotted in Figure 3. For both test sets, The Adaboost classifier performs slightly better than the SVM classifier. Since the SVM classifier is a linear classifier, no polynomial transformations were done on the features, and linear separability in much more unlikely in the feature set, decreased performance would be expected. Another interesting and unexpected result is that the accuracy of the models increases when tested using the secondary test set. This may be indicative of less volatility in the second test set. Lastly, three of the four test cases perform slightly better with fewer final features. This suggest that more features may result in easily overfitting.

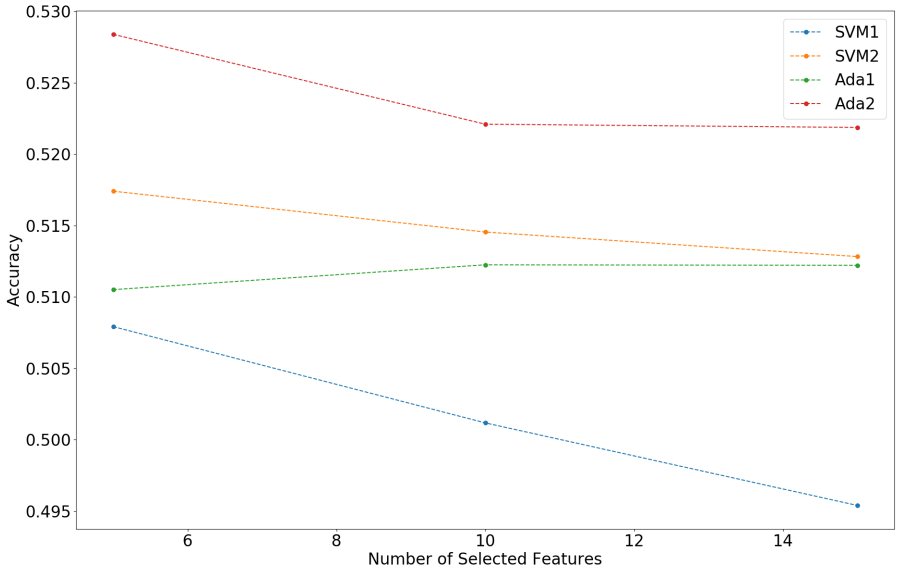

Figure 3: Recursive Feature Elimation Accuracies

3.3 Random Forest Classifier (RFC)

The RFC was calculated with all of the original features and the percentage of feature importances were plotted in Figure 4

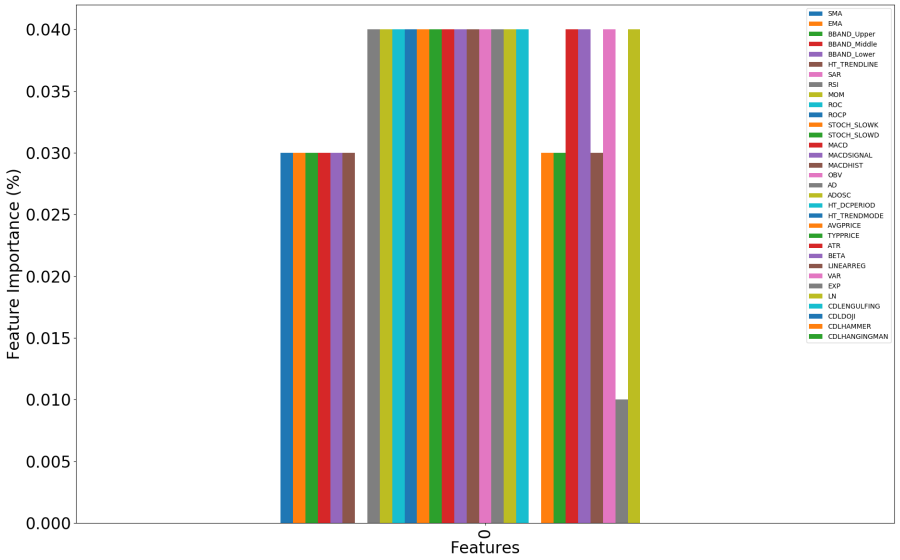

Figure 4: Random Forest Classifier

3.4 Principal Component Analysis (PCA)

Figure 5 depicts the PCA accuracy and time taken with its corresponding reduced feature set of principal components. The results indicate that using fewer principal components (5-10) yields a small, somewhat negligible trade off between the accuracy of the RFC, and the time taken to run the algorithm.

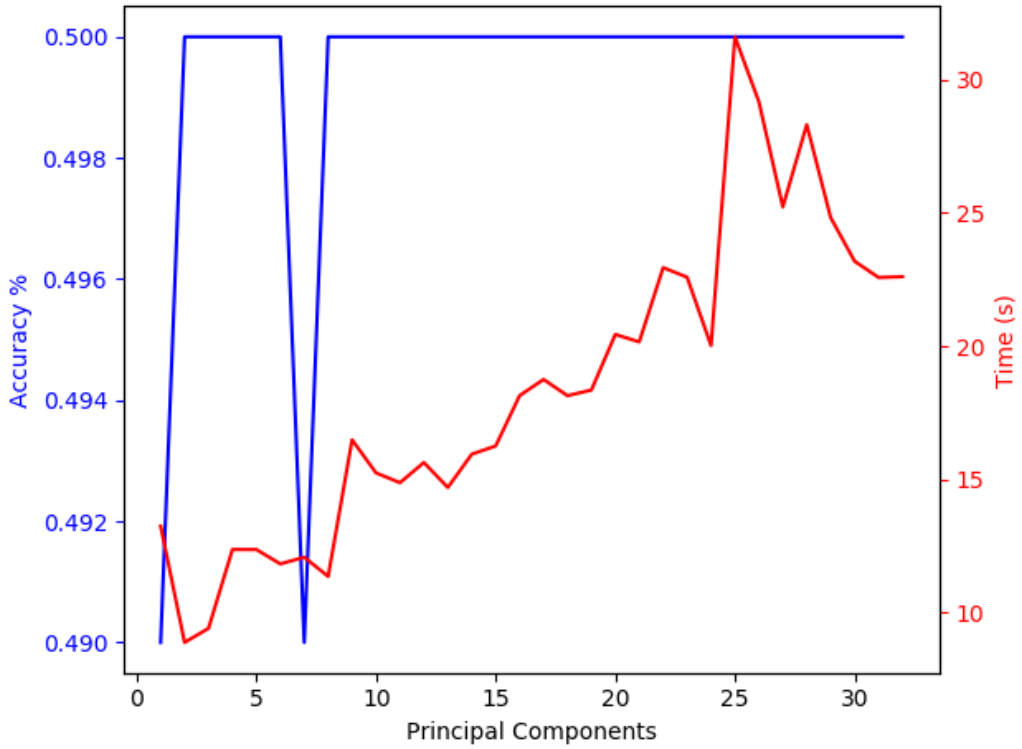

Figure 5: Principal Component Analysis

3.5 Multi Layer Perceptron (MLP)

The results of the three neural nets can be seen in Table 3. Not surprisingly the results are at the same levels as previous methods, however a notable outcome is the handpicked features in the MLP slightly outperforms the other two in the first test set. The MLP with all features has the most accuracy of the three in test set 1 but the least with test set 1. This may point to overfitting similar to the results seen with RFE.

	MLP All Features	MLP Handpicked	MLP RFE-AdaBoost
Test Set 1	0.5068	0.5113	0.5088
Test Set 2	0.5247	0.5174	0.52

Table 3: MLP Results

4 Conclusion

Future work that would yield interesting data would be to look into longer windows in which the label is classified. This paper solely explores a day to day, up or down stock label, and expanding the range of days from 1 day would be a welcomed addition to the quantitative financiers toolkit. Also, we plan on implementing the algorithms in a regression approach that would allow for more

detail and financial flexibility. Having a specific dollar amount and probability of accuracy attached to estimated data would be a desired goal in our future Machine Learning financial endeavors. That being said, or recently incepted hedge fund *We Burn Your Money* or WBYM, is looking for financial backers who are interested in grand profits.

References

[1] David N Reshef, Yakir A Reshef, Hilary K Finucane, Sharon R Grossman, Gilean McVean, Peter J Turnbaugh, Eric S Lander, Michael Mitzenmacher, and Pardis C Sabeti. Detecting novel associations in large data sets. *science*, 334(6062):1518–1524, 2011.

[2] Mahesh Pal. Random forest classifier for remote sensing classification. *International Journal of Remote Sensing*, 26(1):217–222, 2005.