FIN 580: Quantimental Investment and Fintech

Trading Strategy Implementation using Machine Learning Classifiers

Alexander Ilnytsky (ai9)

Habeeb Olawin (holawin2)

Utkarsh Mishra (umishra3)

Professor: Tony Zhang PhD

Advisor: David Wozniak PhD

April 25, 2019

Abstract

The purpose of this paper is to create a robust strategy in predicting the next day returns of equities using simple and advanced machine learning classifiers. We have incorporated technical and sentimental indicators to build our model. In addition to that we have applied a 3-day cutting loss strategy to try to prevent large losses in the event of market downturn.

Introduction

Several papers have analyzed simple and complex machine learning techniques ranging from KNN to hybrid Neural Networks to predict price movements. Most of them have considered technical indicators as inputs to their models. We on the other hand have incorporated both technical and sentimental indicators as inputs to our machine learning classifiers such as Logistic Regression, Support Vector Machine, and Random Forrest in our model analysis.

Exploratory Data Analysis

Using Bloomberg we have selected the 20 largest initial public offerings (IPOs) that went public on New York Stock Exchange (NYSE) or National Association of Securities Dealers Automated Quotations (NASDAQ) with publically traded options from 2014 to 2015. To build our model we pulled historical daily data from January 11th, 2016 to April 16th, 2019 excluding market holidays and weekends. In determining the relationships between selected features we performed exploratory data analysis (EDA) on the data obtained. Our data contains 13 independent features and 817 observations for each stock.

Here are the independent features we used as input to our selected models:

RSI (Relative Strength Index) – 3 Days and 14 Days	Measures the momentum of a security to determine whether it is in an overbought or oversold condition.
EQY_INST_PCT_SH_OUT	Percentage of outstanding shares currently held by institutions.
PUT_CALL_VOLUME_RATIO_CUR_DAY	Ratio of total put volume (volume from all put options, all strikes, and all expirations) to total call volume (volume from all call options, all strikes, and all expirations) traded so far during the current trading day.
PUT_CALL_OPEN_INTEREST_RATIO	Ratio of Total Put Open Interest to Total Call Open Interest
MF_BLCK_1D	Money flow that includes only those trades greater than 10,000 or more shares.
MF_NONBLCK_1D	Money flow that includes only those trades less than 10,000 or more shares.
CMCI	The Commodity Channel Index developed by Donald Lambert, measures the variation of a security's price from its statistical mean. A CMCI indicator falling below a value of 100 indicates an oversold condition. Similarly, a CCI value greater than 100 indicates an overbought condition.
FEAR_GREED	Fear/Greed is an oscillator based on the Average True Range (daily high/low range, adjusted for gaps) to measure the ratio of buying strength to selling strength. This tells us whether the Bulls or the Bears are in control at a particular

	point in time.
MACD_DIFF	An indicator of the change in a security's underlying price trend. The theory suggests that when a price is trending, it is expected, from time to time, that speculative forces "test" the trend. MACD shows characteristics of both a trending indicator and an oscillator.
volume%	Volume to Shares Outstanding ratio during the current trading day
IVOL_MONEYNESS	Implied Volatility using 100% Moneyness
PCT_INSIDER_SHARES_OUT	Percentage of outstanding shares currently held by insiders.

Why these Features?

RSI_3D and RSI_14D: These attributes measure the momentum of a security to determine whether it is in an overbought or oversold condition. They will tell us the direction in which the stock price would go the next day just by looking at the values of the features. If RSI is above 70, the stock is likely to pull back in the near future. In turn, if RSI is below 30, the stock price will likely go up the next day.

Implied Volatility by Moneyness: If implied volatility is high, it indicates large stock price movements in the future.

Percentage of Shares Outstanding held by Institutions and Insiders: Usually institutional investors and company employees have more knowledge about the future prospects and conditions of a firm rather than retail investors. Hence these features can tell us whether the stock will go up or down

Put-Call Volume Ratio of Current Day: This feature tells us what speculators believe the direction of the stock will be in the future.

Moneyflow Block and non-Block 1 Day: Reports the money flow (positive or negative) for a particular stock. Positive money flow indicates that prices are likely to move higher, while negative money flow suggests prices are about to fall.

Commodity Channel Index: This technical indicator will tell us whether the stock was overbought or oversold. If the indicator is less than -100 then the security is oversold indicating the price will go up, otherwise if indicator level is greater than 100 then the stock is overbought indicating the price will go down. This indicator can confirm the signal of the RSI prediction.

Fear Greed: This indicator tells us whether the bulls or the bears have control of the stock. If the bulls have control, the price is likely to go up otherwise when the bears dominate, the price is likely to go down.

MACD Difference: If the value is above 0 then it's a buy signal i.e. the price is going to go up but if the level is below 0 then it's a sell-time i.e. price will go down

Volume%: The higher the indicator's level higher the trading volume for that particular day. It does not tell us about the direction of the stock price

Dependent Variable

The dependent variable in our trading strategy is the next day return. Since our model utilizes a machine learning classifier, we apply the following rule - if the price goes up the next day then we assign a value of 1 to the next_day_return, else we assign next_day_return a value of 0.

One of the EDA techniques is to look at the heat map to determine correlations between the features. Below are the heat maps we produced for Alibaba and Synchrony Financial stocks in figures 1 and 2.

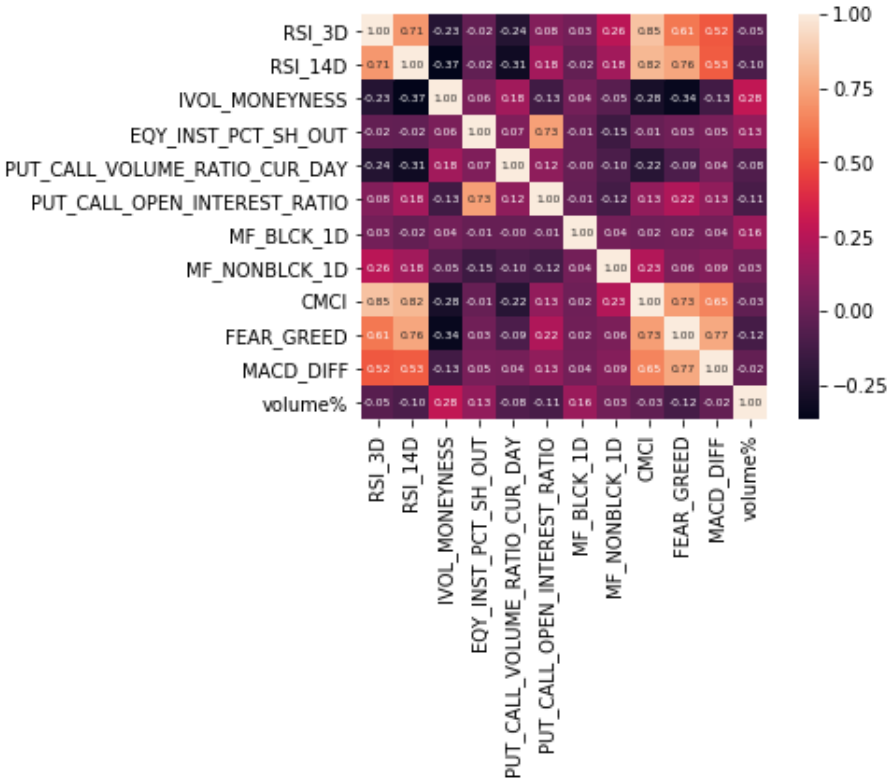

Fig 1. Alibaba heat map

We can observe that MACD_DIFF and FEAR_GREED has the positive highest correlation while IVOL_MONEYNESS and RSI_14D have the highest negative correlation amongst the features for Alibaba stock. It shows that security's underlying price trend is driven by market attitude (bullish or bearish) towards the stock. On the other hand, implied volatility does not tell us whether it is an overbought or oversold condition for the stock.

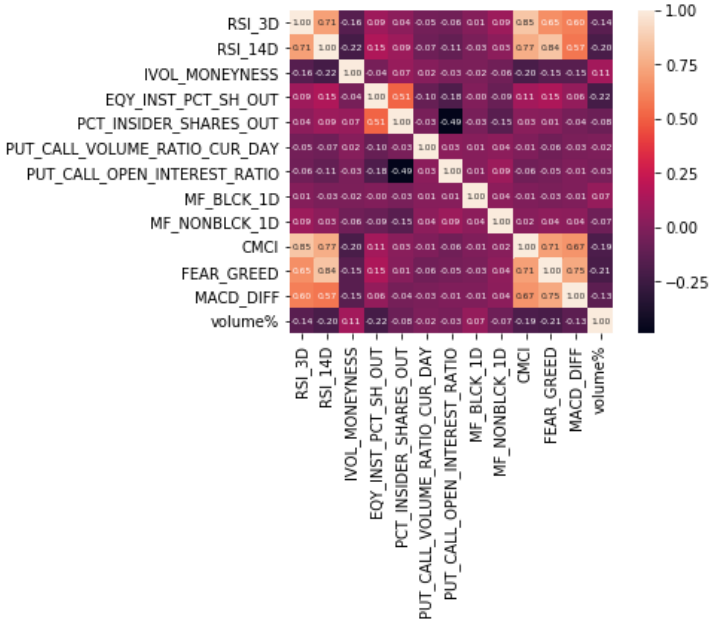

Fig 2. Synchrony Financial heat map

For Synchrony Financial, CMCI and RSI_14D have the highest positive correlation whereas PUT_CALL_OPEN_INTEREST_RATIO and PCT_INSIDER_SHARES_OUT have the highest negative correlation.

Pre-processing

We have split our data into 80% training (652 observations) and 20% test set (163 observations). To be more specific, our model gets trained on data corresponding to January 11th, 2016 to August 17th, 2018 with a holding period from August 20th, 2018 to April 16th, 2019.

While pre-processing, we checked and removed missing values and also calculated a new independent variable - Volume% to be used in our model. Next, we standardized the features to bring attribute values on the same scale with zero mean and unit variance.

To select the relevant features from the dataset, we used a random forest learning method because of its relatively good accuracy, robustness, and ease of use. We found the feature importance of the dataset via the feature_importances function after fitting a RandomForestClassifier. We demonstrate feature importance for 3 stocks – Synchrony Financial, JD, and Santander Consumer USA due to space constraints as seen in figures 3, 4, and 5.

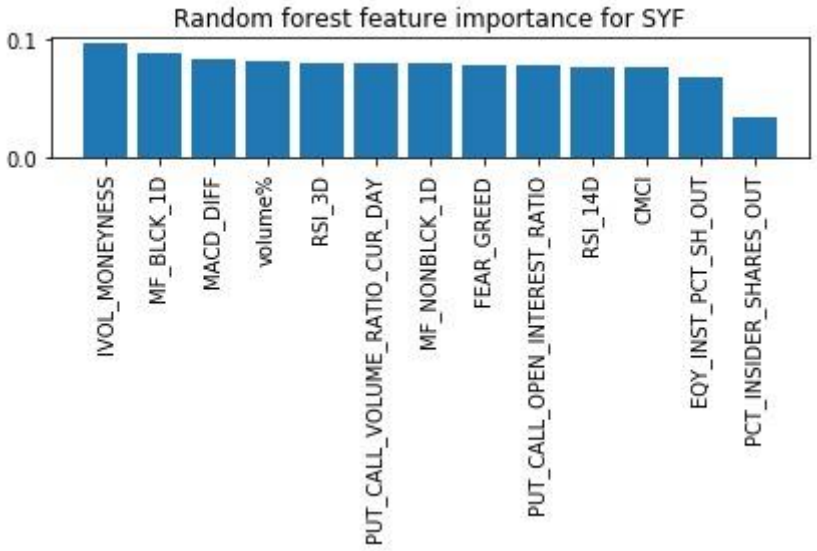

Fig 3. Synchrony Financial feature importance

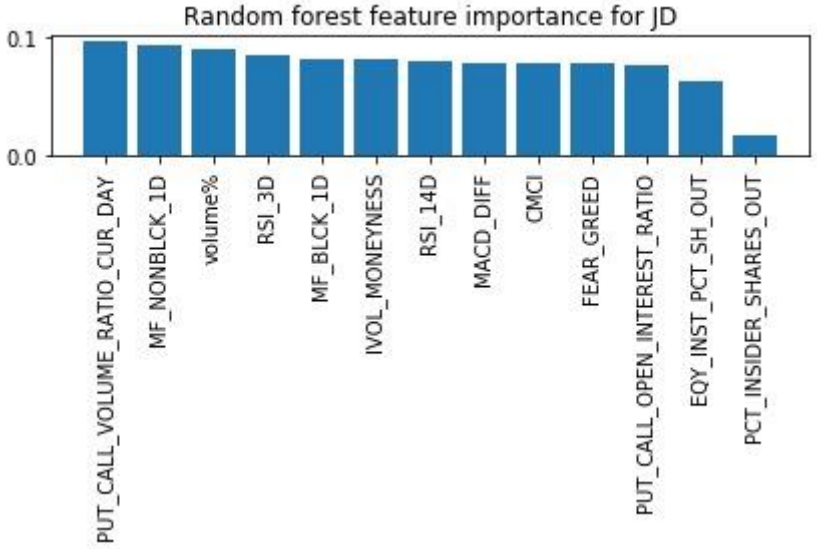

Fig 4. JD feature importance

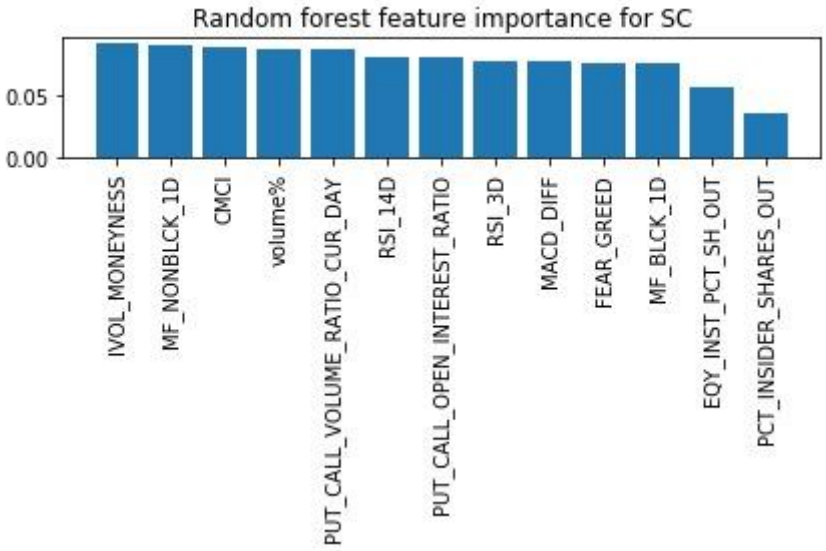

Fig 5. Santander Consumer USA Holding Inc feature importance

Finally, to summarize the feature importance of our dataset, we transformed it onto new feature subspace of lower dimensionalities compared to the original one. For our project, we performed principal component analysis (PCA) feature transformation technique to find the direction of maximum variance in the data and project it onto a new subspace with equal or fewer dimensions than the original one. It was essential to standardize the features prior to PCA transformation because PCA dimensionality reduction technique is highly sensitive to data scaling and we wanted to ensure that we were assigning equal importance to all features.

Strategy

First we categorized next day returns into 0s and 1s. Then we predicted the direction of next day returns using selected machine learning classifiers. If the model predicted 1 then we take or keep long position in the stock. Otherwise, if model predicted 0 we close the long position.

Moreover, we have also considered cutting losses by assigning a trigger event – if log returns of a particular stock are negative for three consecutive days; we close our long position in the stock.

The main objective of our strategy is to predict whether the next day return will be positive or negative. When our model predicts that the next day return is positive we open a position in this security at the end of today, if the next prediction is still positive, we hold on to our position. Finally, if the next day return is predicted to be negative we exit our long position and we will not re-open the position until we get a positive prediction.

Model Fitting and Evaluation

Three different machine learning models have been applied to the dataset to classify the direction of log returns of the portfolio as well as of individual stocks: Logistic Regression, Support Vector Machine, and Random Forrest Classifier. Following are the accuracy scores:

Model	PORTFOLIO	BABA	SYF	JD	SC	CFG
Logistic Regression on PCA (10) transformed data Training Data Accuracy	0.5482	0.5537	0.5429	0.5276	0.5399	0.5706
Logistic Regression on PCA (10) transformed data Test Data Accuracy	0.5085	0.5427	0.5366	0.5305	0.5061	0.5244
Logistic Regression on PCA (10) transformed data Test Data Accuracy with Cutting Loss Strategy	0.5140	0.5488	0.5488	0.5305	0.4817	0.5976
Support Vector Machine on PCA (10) transformed data Training Data Accuracy	0.6623	0.6840	0.6871	0.6733	0.6718	0.6457
Support Vector Machine on PCA (10) transformed data Test Data Accuracy	0.5034	0.5610	0.4512	0.5488	0.5122	0.4817
Support Vector Machine on PCA (10) transformed data Test Data Accuracy with Cutting Loss Strategy	0.5061	0.5671	0.4512	0.5366	0.4878	0.5610
Random Forrest on PCA (12) transformed data Training Data Accuracy	0.6445	0.6012	0.6488	0.6672	0.6273	0.6350
Random Forrest on PCA (12) transformed data Test Data Accuracy	0.5125	0.5488	0.5122	0.5671	0.5610	0.5061
Random Forrest on PCA (12) transformed data Test Data Accuracy with Cutting Loss Strategy	0.5155	0.5488	0.5183	0.5305	0.5305	0.5610

Model	PE	AXTA	TGE	INFO	FIT	AY
Logistic Regression on PCA (10) transformed data Training Data Accuracy	0.5383	0.5322	0.5276	0.5552	0.5506	0.5675
Logistic Regression on PCA (10) transformed data Test Data Accuracy	0.5244	0.5305	0.4817	0.5122	0.5244	0.4878
Logistic Regression on PCA (10) transformed data Test Data Accuracy with Cutting Loss Strategy	0.5427	0.5305	0.4390	0.5183	0.4939	0.4878
Support Vector Machine on PCA (10) transformed data Training Data Accuracy	0.6871	0.6672	0.6304	0.6227	0.6856	0.6764
Support Vector Machine on PCA (10) transformed data Test Data Accuracy	0.4695	0.5122	0.4695	0.5244	0.5122	0.5061
Support Vector Machine on PCA (10) transformed data Test Data Accuracy with Cutting Loss Strategy	0.5061	0.5122	0.4207	0.5305	0.4939	0.5000
Random Forrest on PCA (12) transformed data Training Data Accuracy	0.6503	0.6733	0.6948	0.6242	0.6350	0.6794
Random Forrest on PCA (12) transformed data Test Data Accuracy	0.5427	0.5183	0.4634	0.4573	0.4878	0.4878
Random Forrest on PCA (12) transformed data Test Data Accuracy with Cutting Loss Strategy	0.5610	0.5244	0.4146	0.4634	0.4817	0.4817

Model	MR	SHLX	CTLT	LC US	RACE US	TRU US	SABR US	KRNY US	SERV US
Logistic Regression on PCA (10) transformed data Training Data Accuracy	0.5567	0.5521	0.5199	0.5506	0.5613	0.5583	0.5429	0.5613	0.5552
Logistic Regression on PCA (10) transformed data Test Data Accuracy	0.5122	0.5671	0.4878	0.5000	0.4634	0.4878	0.4634	0.5122	0.4756
Logistic Regression on PCA (10) transformed data Test Data Accuracy with Cutting Loss Strategy	0.5305	0.5610	0.5488	0.5000	0.4573	0.4756	0.4878	0.5122	0.4878
Support Vector Machine on PCA (10) transformed data Training Data Accuracy	0.6442	0.6748	0.6549	0.6426	0.6641	0.6534	0.6794	0.6426	0.6595
Support Vector Machine on PCA (10) transformed data Test Data Accuracy	0.5427	0.4695	0.5976	0.4817	0.4817	0.5122	0.4634	0.4756	0.4939
Support Vector Machine on PCA (10) transformed data Test Data Accuracy with Cutting Loss Strategy	0.5793	0.4573	0.5976	0.4939	0.4634	0.5000	0.4878	0.4756	0.5000
Random Forrest on PCA (12) transformed data Training Data Accuracy	0.6365	0.6917	0.6610	0.5951	0.6672	0.6074	0.6334	0.6518	0.6089
Random Forrest on PCA (12) transformed data Test Data Accuracy	0.5549	0.5427	0.5000	0.5061	0.4939	0.4878	0.4878	0.5061	0.5183
Random Forrest on PCA (12) transformed data Test Data Accuracy with Cutting Loss Strategy	0.5915	0.5366	0.5549	0.5122	0.4756	0.4756	0.5122	0.5061	0.5305

To get the best results, we have fine-tuned our model to determine that 10 components for PCA transformation gave the best accuracy for Logistic Regression and Support Vector Machine classifiers. In the case of Random Forrest, we transformed our data using 12 components of PCA transformation technique after fine-tuning to get the best accuracy.

We have also calculated mean, standard deviation and value at risk (VaR) for our portfolio and strategies using the 3-day cutting loss strategy and without the 3-day cutting loss strategy.

Results

In the first model, we tested Logistic Regression with a PCA (10) transformation as seen in figure 6 below.

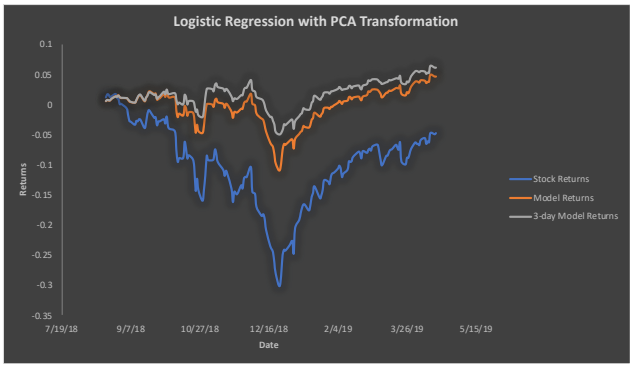

Fig 6. Logistic Regression with PCA transformation

This model outperforms the long only (Stock Returns) portfolio during the testing period. Holding the portfolio yields us a -4.69% return while our model generates 4.7% return. Additionally, the day to day fluctuations for our model are lower which is expected. The maximum possible total loss we observed by holding the portfolio was -30.05% at the end of December 24, 2018, whereas that loss was -10.88% for our model. By adding the stop-loss condition we were able to increase the total return to 6.15%, further reducing the day to day volatility and decreasing the maximum possible total loss to -4.9%.

Next, we look at the performance of Support Vector Machine Model (SVM) as seen in figure 7 below.

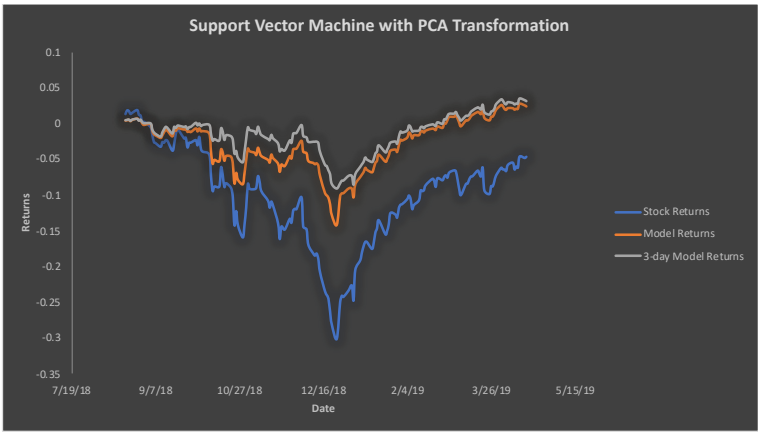

Fig 7. Support Vector Machine with PCA transformation

This model also outperforms the portfolio during the testing period. SVM generates 2.45% return during the holding period. The day to day fluctuations for this model are also lower. The maximum possible total loss we observed during the holding period for SVM model was -14.19% at the end of December 24, 2018. By adding the stop-loss condition we were able to increase the total return to 3.12%, while reducing the day to day volatility and decreasing the maximum possible total loss to -9.1%.

Lastly, we will finally apply the Random Forest model as seen in figure 8 below.

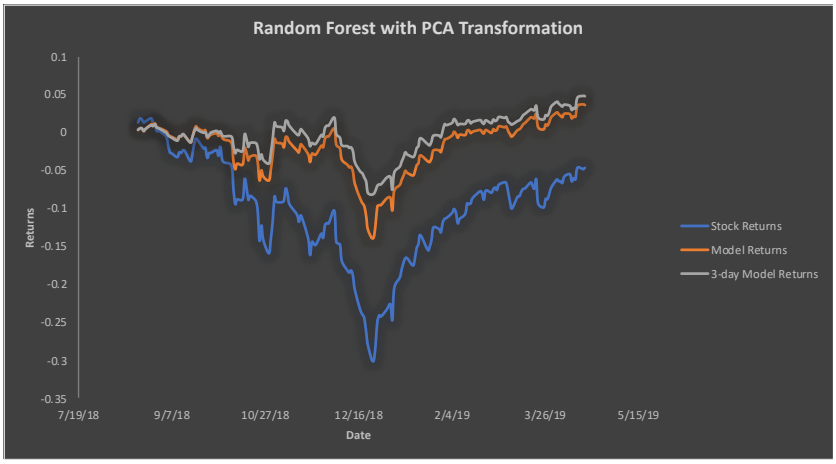

Fig 8. Random Forest with PCA transformation

We concluded that the Random forest (RF) classifier performs best with following parameters: criterion='gini', max_features =12, max_depth = 2, n_estimators=1000. This model also outperforms the portfolio during the testing period. RF generates 3.6% return during the holding period. The day to day fluctuations for our model are also lower. The maximum possible total loss we observed during the holding period for our RF model was -13.86% at the end of December 24, 2018. By adding the stop-loss condition we were able to increase the total return to 4.70%, while reducing the day to day volatility and decreasing the maximum possible total loss to -8.07%.

Finally, we will compare the performance of our models against each other. Surprisingly, the Random Forest model is not the best performing one. Logistic Regression on PCA transformed data with a stop-loss condition gave us the best return and the lowest variance with the lowest 95% VaR (Table 1).

Fig 9. Model Comparison

Daily Returns		
Portfolio		
95% VaR	Mean	S.D
-2.15%	-0.06%	1.36%
Logistic PCA(10) 3-day		
95% VaR	Mean	S.D
-1.03%	0.04%	0.59%
Logistic PCA(10) without 3 day		
95% VaR	Mean	S.D
-1.26%	0.03%	0.76%
SVM PCA(10) with 3 day		
95% VaR	Mean	S.D
1.11%	0.02%	0.64%
SVM PCA(10) without 3 day		
95% VaR	Mean	S.D
-1.29%	0.02%	0.82%
Random Forrest PCA(12) with 3 day		
95% VaR	Mean	S.D
-1.08%	0.03%	0.69%
Random Forrest PCA(12) without 3 day		
95% VaR	Mean	S.D
-1.46%	0.02%	0.86%

Table 1. Daily VaR (1-day), mean, volatility (Standard Deviation) of portfolio

Model Limitations:

- The first limitation is that we are not accounting for the transaction costs. At the absolute worst-case scenario, assuming 5$ per trade commission, we will spend $5*20*163=$16300 in total. However, the real commission will be significantly smaller, as we are not realistically exiting positions every day. With amount invested being large enough, this cost will be insignificant.
- We are testing our data on a fixed period. We cannot guarantee that our model would perform the same in a different time period.
- We can only apply our strategy to stocks which have traded options. The model should not be used with stocks with no traded options.
- We need stocks with enough trading history, in order to train our models for best performance.
- The stop-loss condition worked well during the selected time frame due to the market downturn, but may not work as well during the period of low volatility.

Conclusion:

The purpose of our model is to use the available stock information to predict the direction of price movement on the next day. To achieve this we applied different transformation techniques on our dataset before training our classifier model and observed best results with PCA transformed standardized data. We have applied several classifiers, including Logistic Regression, SVM, and Random Forests. For all 3 models we were able to achieve greater than 50% accuracy. This model can be extended to all securities with enough trading history with options data and not just the IPOs we have analyzed.