Udacity Machine Learning Capstone Project, Build a Stock Trade Adviser **Patrick Parker, 10-29-19**

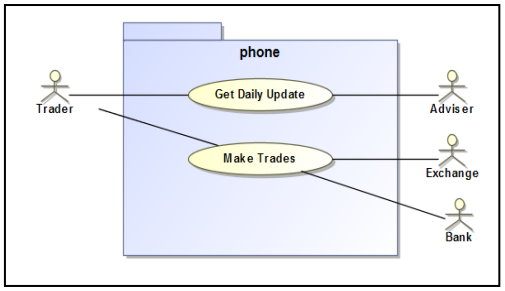
Executive Summary: In this project, a machine learning based stock adviser is built. Sharpe ratio's of ~1.7 were found and beat the market percentages are largely positive over 10%. Using good quality stocks, a historical shape indicator of reasonable performance was devised, and a trade classifier was trained to improve the odds of placing a good trade. Together these all afford good stock trade advice, deployed in the palm of the phone via an AWS S3 website. This stock trade adviser is intended to complement phone-based stock exchange activity on the fly.

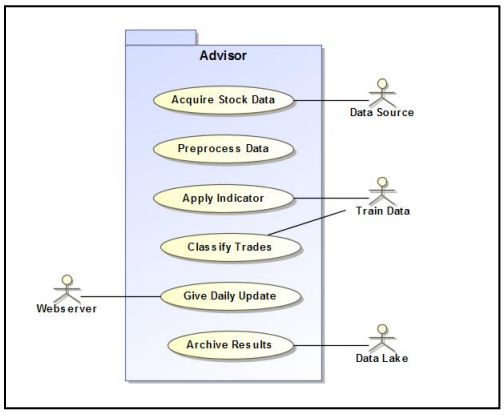
Introduction: This project is intended for "small account part time mobile phone traders". They use a stock exchange website where they buy and sell when they have a spare moment. This is a long trading strategy where they buy first then sell in small amounts. The time frame is weeks to months, not day trading with real-time updates; it's simply daily updates. This project is not a stock exchange, but aims to help in this. Simply, it's a phone accessible stock trade adviser that aims to provide timely information. This doesn't eliminate bad trades, though it may help to make more good trades than bad.

Disclaimer: The application of this project to real world trading is not recommended as results, trading objectives, and trading styles significantly vary.

Project Overview: This project is primarily an educational exercise in Machine Learning. SysML diagrams depict what it involves and covers use cases, activities, package, and requirements.

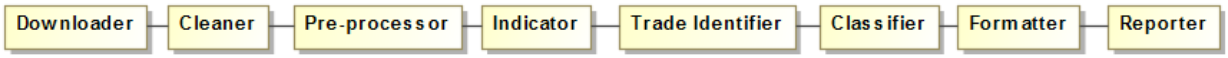
Phone Use Case: The phone does a lot of things, but here it's about trading. A trader uses the phone to get advice and make trades. The project plays the role of the Adviser in this scenario. The Trader first gets daily update, decides on what trades to make, then proceeds to make trades. The Adviser provides the daily update, the exchange and bank provide means to make trades.





Adviser Use Case: The Adviser incorporates a variety of things supporting daily update generation: downloads stock data, machine learns it, and hands-off good advice to the Webserver that conveys it to the Trader's phone. In simple terms, this is an Extract, Transform, Load worker.

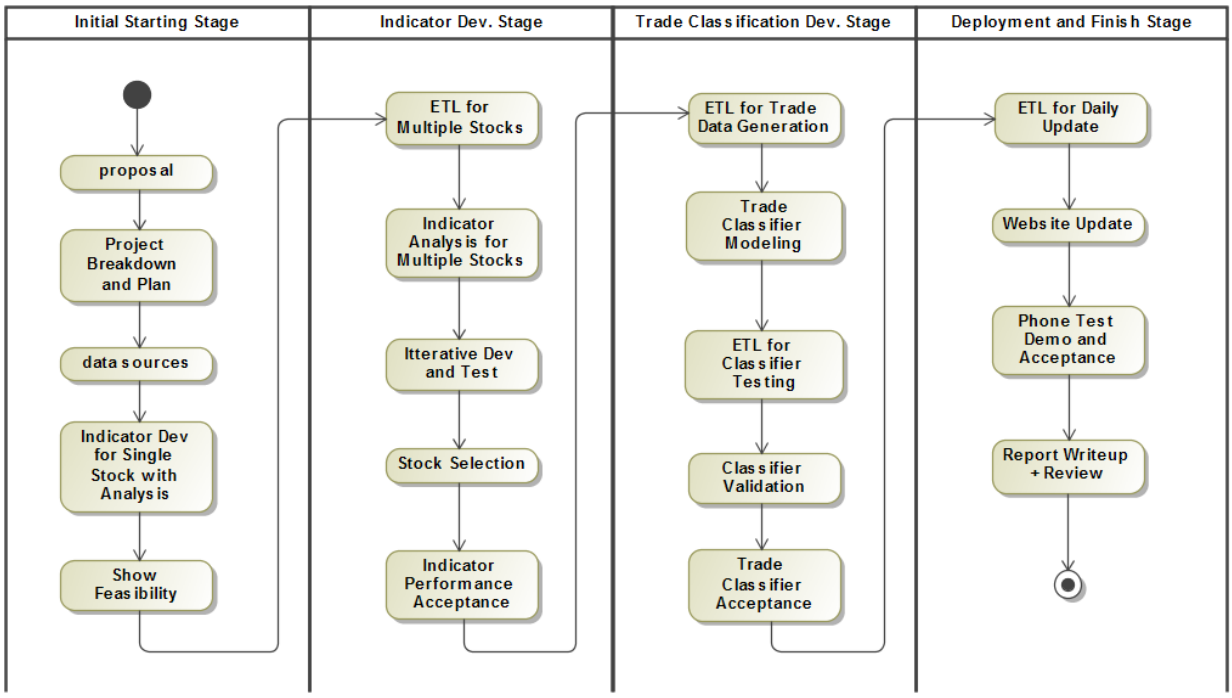
Model Architecture: It's a sequential flow pipeline of data from left to right using an ETL approach:



The Adviser operates as an ETL process composed of eight basic building blocks.

1) The downloader extracts data from the data source.
   In this case the data is adjusted close which incorporates stock splits and dividends.
2) The Cleaner starts the data transformation. It conditions the data for preprocessing.
   The adjusted close is used to appropriately scale Open, High, Low, Close features.
3) The preprocessor converts market price versus time movements into shape features.
4) The indicator converts shape features into a timewise varying indicator trace.
   Note the indicator here is custom trained on historical data for specific markets.
5) The trade identifier compares the indicator to threshold levels to build trade signals.
   Likewise, the threshold levels are custom trained on historical data for specific markets.
6) The classifier assesses trade signals for probability of being a high return trade.
   This classifier – trained on trading data from the group of markets studied.
   Trade opportunities are compiled for downstream reporting.
7) The formatter rank sorts the trade opportunities based on probability.
   The probability relates to the likelihood of making a high return trade.
   These probabilities are based on historical data from the group of markets studied.
8) The reporter loads ranked opportunities to the webserver.
   This isn't a real-time adviser, daily updates suffice.

Project Workflow Activity: The project is split up into stages that are approached sequentially. As problems arise, the stages are iteratively refactored until the stage workflows work congruently.

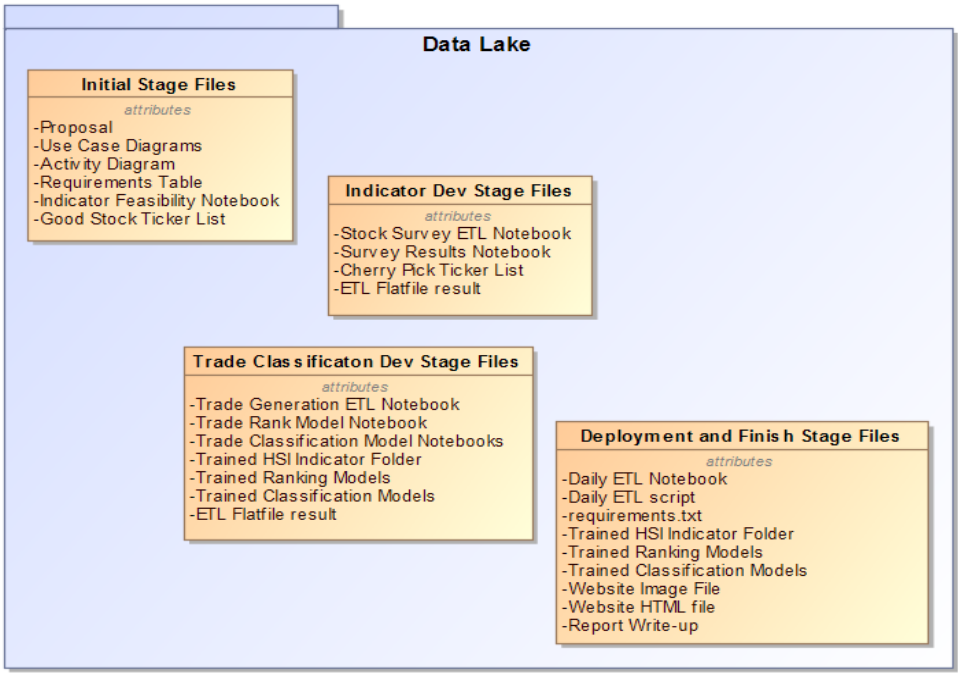


Project Requirements: A little time up front here pays dividends later.

At the project finish this acts as a double-check list to verify it's an okay success.

#	Type	Requirement Description	Justification Note
1	Programming Language	Project shall be written in Python 3.x.	Python 2 phasing out.
2	Software Libraries	Information regarding software libraries shall be included in readme file.	Environment dependencies.
3	Data Accessibility	Data shall be publicly accessible.	Openly shared project.
4	Data Ethics	Unethical choices of data shall be avoided.	Trade Disclaimers prominently included.
5	Submission Files	Proposal, report writeup, notebooks including test results and sample data shall be included in review submission.	GitHub compatible.
6	Model Architecture	Description of the model architecture shall be included in report.	Communication of core ideas.
7	Report Test Results	Test results shall be discussed including model accuracy figures.	Basic functional performance.
8	Report Improvements	Report shall include three areas for future improvement.	Iterate for ongoing improvement.
9	Mobile Phone Access	Model results shall be accessible by mobile phone.	Basic front-end function for data access.
10	Model Updates	Adviser recommendations shall update daily using pre-trained indicator and classification models.	Basic function for timewise advice.
11	Stock Tickers	Stock tickers shall be cherry-picked for the best chance of beating the market.	Avoid GIGO effect to start with.
12	Market Pre-test	Market models that pretest poorly shall be excluded from study.	Downstream GIGO check.
13	Data Pipeline	Data shall flow in an Extract Transform Load (ETL) data pipeline.	Basic back end function.
14	Data Lake	Flat file artifacts shall be stored in a well-organized data lake.	Support project sustainability.
15	Indicator Sharpe Ratio	A minimum indicator Sharpe Ratio of 1 shall be used as the acceptance threshold.	Performance essential for good advice.
16	Classifier Accuracy	A trade quality classifier accuracy above 60% is acceptable.	Well above 50% is considered an edge.
17	Statistical Testing	A t-test comparing indicator mean with random return means within three years shall indicate significance.	Weak trend trades rely most on indicator.
18	Benchmark Comparison	Accepted indicators shall beat the market, as compared with the S&P500 index in similar time periods.	Stretch goal for good advice purposes.
19	Project Completion	Project shall be completed on-time or earlier.	Best known practice.
20	Cloud Architecture	Project shall utilize AWS S3 infrastructure.	Best known utility.

Project Data Lake Package Diagram: The organization follows the activity stages. A data lake approach is adopted to provide flexibility to flat file generation as the project develops. The folder attributes depicted generally describe where project components reside.

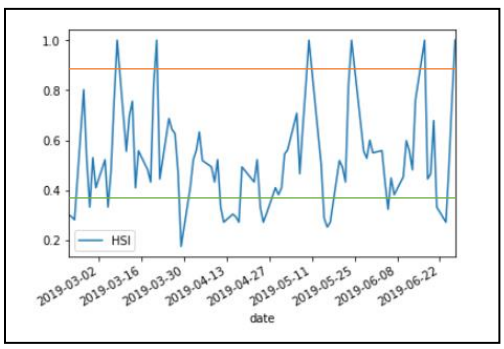


Project Data: Raw data for this project is open source historical stock data from the US stock market.

There are many stocks to choose from. To better the chances of a success, high quality stocks are considered, and there's many opinions about what defines a high-quality stock. I don't know. For this reason, "The Street" recommended stocks were considered from the basis that this is a long-time stock adviser site who's survived the test of time. In this approach, "The Street" quality criterions are a fallback when things go awry. Stock symbol "tickers" were retrieved from "The Street" website. Then historical data for those tickers was mined using an "Alpha Vantage" data mining python package. Mined data includes "Open, High, Low, Close, Volume; Dividend, Split, Adjusted Close". Initially over 100 stock tickers were data-mined over the last ten years. Ultimately a large fraction of these were selected as working acceptably with the indicator developed.

Starting Stage - Indicator Development

The Historical Shape Indicator (HSI): The Trader needs to know when to buy or sell. The Indicator is a timewise trace that crosses two horizontal threshold levels. Upon those crossings, trading signals are triggered to buy or sell. The Historical Shape Indicator picks up shape features from timewise price fluctuation features.



Historical Shape Ordinals: If the market goes down then back up, we may call that a "V" shape, but markets can make more shapes then letters in the alphabet. For this reason, a numerical Ordinal data type is chosen. Ordinals are categorical data types; here they are assigned to shape features. During training, the ordinals are assigned look forward return figures. For example, when a market makes a 'V' or 'W' shape, we know it goes up some percent in the near future and so we might buy on that shape; whereas an 'M' shape may likely go down in the near future – we might sell before that happens. Not only is there variation in shapes, there's variation in look-forward returns, specific to the market's personality. This is why the Historical Shape Indicator is customized to the market it is used with. Prior historical data is used to train it for use in predicting future price movements of the same market.

Building Day Shape Ordinals: To convert "Open, High, Low, Close" into an ordinal, a simple approach is taken using average and standard deviation figures for those features. Specifically, the difference in their average from the day before is compared to the standard deviation. If the difference in average is larger than a standard deviation, as in the market going up, it is assigned an "up" character, for example a "3". If the difference in average is within one standard deviation, as in the market remaining flat, it is assigned a "flat" character, for example a "2". If the difference in average is less than a negative standard deviation, as in the market going down, it is assigned a "down" character, for example a "1". These characters are ordinals or categorical number types representing the day shape.

Building Shape Ordinals: The day shape is a building block for longer span shapes because the day shapes can be chained together. To form a 5-day shape ordinal, "11233" might symbolize a 'V' shape: where price went down two days (represented by two '1's), flat one day (represented by a 2), and up two days (represented by two '3's.) With a 5-day ordinal possibility, there's 3^5 combinations or 243 distinct shapes, (and this is only one possible shape arrangement.)

Ranking Shape Ordinals: Once the ordinals are assigned, then for each day the average look forward returns are computed. In this study, mean 7-day look forward returns are used, (and again this is only one possibly arrangement.) To get the mean look forward returns, we first shift daily log returns by 7 days back from the future. Then we average together all the shifted returns that occurred on days sharing the same shape ordinal. Next, the set of all shapes is then ranked from high to low by these look forward mean returns.

Ranked Ordinals to HSI: The mean look-forward return figures likely go from large positive to large negative values depending upon the market personality. So, these are normalized between 0 and 1 and this forms a Historical Shape Indicator (HSI). The trained indicator model then, is simply a shape to HSI lookup table. Knowing all this, we understand when a certain shape gets made and the HSI indicated is high, the market has historically been ripe for high look forward returns. Likewise, other shapes maybe indicate the opposite happening when their indicated HSI goes low, we expect the market to fall.

Tuned Threshold Crossings: Optimal threshold crossings are formed by back testing over history. Basically, using successive approximation, these thresholds are moved until their resulting aggregated trade returns are maximized, given trading rules that say "When HSI crosses above the High-Threshold line, we buy; when HSI crosses below the Low-Threshold line, we sell."

Test Train Splits: We cannot test performance using the training data, because that's misleading. There's a careful test/train split approach used where years prior get used to train for a year post testing. For example, to train the HSI for the year 2018, the years 2015,2016,2017 may be used for training. Performance in 2018 is based on HSI generated trades using price data it had never seen, the test data is distinctly after the historical data that the HSI was trained upon.

Historical Shape Composite (HSC): Other time spans beyond 5 days are combined into an ensemble composite, Historical Shape Composite or HSC. Here 3- and 5-day spans were used for the average and standard deviation day shape analogy. A majority vote is taken between three HSI lookups (1-day,3-day,5-day) to form the HSC. Note that the 5-day span shape ordinals stretch over 25 days past, it's a slow shape indicator balancing the faster shapes.

Programming HSC: More feature columns are built side by side with each, but they depend on row history. For this reason, leading rows cannot be filled out completely because there's no history for them. In these cases, these rows are dropped after the build. Price estimates (pe1, pe3,pe5), standard deviations (sd1,sd3,sd5), day shapes (ds1,ds3,ds5), shape ordinals (shp1,shp3,shp5), historical shape indicators (hsi1,hsi3,hsi5), and HSC then all become market features based upon recent market shape history, each within themselves reflecting differing time spans. For example, pe1 is a price estimate for a 1-day time whose shape ordinal shp1 spans five days, whereas pe3 is a price estimate for 3-day time span whose shape ordinal shp3 spans 15 days.

Indicator Notebook: An example of indicator development is Jupyter notebook is "Visa_shape.ipynb." This contains data download routines, preprocessing, basic returns, shift returns, shape assignments, etc. it likewise contains statistical T-tests, Sharpe Ratio and Beat the Market figures that seem encouraging (keep in mind that Visa is a strong bull market, so most things look good!) Notebooks like these were made for various single markets before the stage milestone of showing Indicator feasibility was accepted.

Indicator Notebook Results: Results of this initial study look promising with Sharpe ratios>1 in most cases, and positive expected returns. The volatility is less than 20% and it beats the market.

Note the p-value fluctuates. This may indicate the fallback condition where the market is trending to such a degree, that with or without the indicator it would do well.

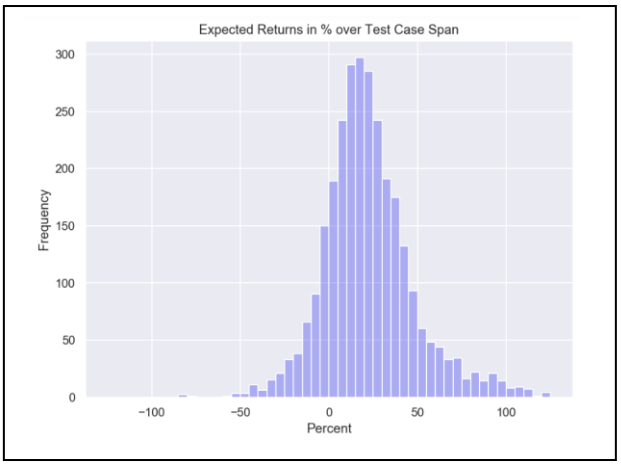
symbol	equity	test_year	price$	lowT	highT	#trades	in_days	t-val	p-val	exp_ret%	volatility	years	sharpe	beat%
V	Visa	2011	24.140825	0.13750	0.67500	14.0	200	0.052610	0.958165	35.073489	0.172597	0.989041	1.851838	30.369757
V	Visa	2012	35.651762	0.30000	0.75625	19.0	100	1.622776	0.106339	33.594293	0.111665	0.994521	2.711623	25.669118
V	Visa	2013	53.184720	0.21875	0.74375	14.0	163	2.470578	0.014451	52.054075	0.128628	0.994521	3.797073	28.582710
V	Visa	2014	63.794290	0.18125	0.68750	15.0	134	-0.023865	0.980980	9.034544	0.134732	0.994521	0.414477	4.014917
V	Visa	2015	75.876887	0.30625	0.76250	14.0	96	0.388587	0.698070	16.725242	0.150828	0.994521	0.882955	21.426308
V	Visa	2016	76.803595	0.29375	0.55000	33.0	169	1.916904	0.057164	17.284696	0.131720	0.989041	1.061052	8.448825
V	Visa	2017	113.127180	0.28750	0.76875	15.0	153	2.860153	0.004721	34.104995	0.086184	0.986301	3.606081	21.774856
V	Visa	2018	130.950964	0.29375	0.73125	13.0	188	-1.792506	0.075684	5.400655	0.173565	0.994521	0.111221	13.331135
V	Visa	2019	176.590000	0.25625	0.71250	15.0	127	-0.434630	0.664466	22.483021	0.121174	0.791781	2.054520	16.486876

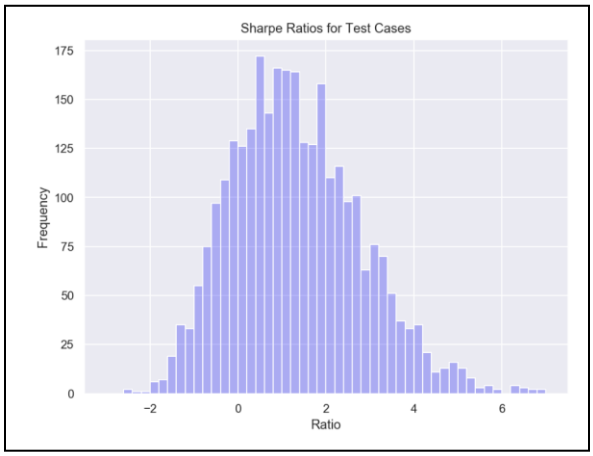
Second Stage – Further Indicator Development

Multiple Stock Survey: After demonstrating feasibility, one or two stocks is not enough because we want to offer stock advice for multiple opportunities. Here, an ETL process runs stock choices from "The Street"(ref.1); to see how well it works on a variety of stocks. This consumes many hours because there's many iterations. Approximately 185 stocks were surveyed using the indicator. The test years spanned from 2002 to 2019, if that stock data was available, using the two-years prior for training in each case. This ETL generates a results data-frame that includes things like Stock Price, HSC trade thresholds, trades per year, days in market, days per trade, p-values, expected returns, trade volatility, Sharpe ratio, and Market Benchmark used to visualize ~3000 case studies in a separate analysis notebook.

Multiple Stock Results: This notebook is called 'multiple_stock_analysis.ipynb', it imports the ETL results and builds visualizations of them. There are 2952 test cases from 185 stock tickers. The average stock price is $50. The average number of trades per year is 17. The average number of days per trade is 10.

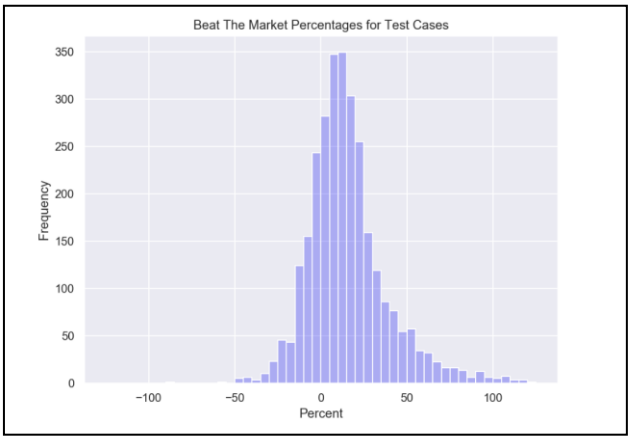
The mean of the aggregated expected returns is 18%. Expected returns come from the daily returns over the test span, typically 1 year.





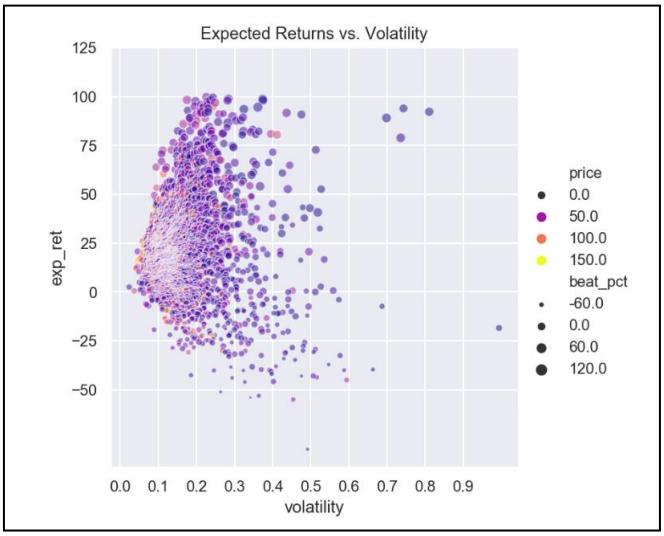
Likewise the mean Sharpe Ratio is 1.33 lending hope in having some good, as greater than 1.

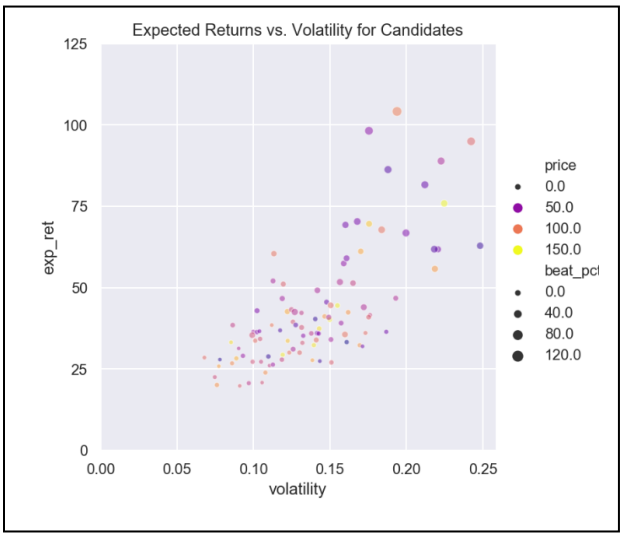
And furthermore, the beat the market percentage seems cooperative with a mean of 15.5%.



Stock Selection: The basic assessment looks promising, however the negative portions seem undesireable. There's some likelihood that some stock tickers complement this approach more than others. If done well, with good stock selection, the negative portions may diminish. To do this, the Sharpe ratio is prefered because it factors volatility risks. A study of optimization here reveals the "efficiency frontier", where the top edge of a returns vs. volatility plot - indicate cases where the highest return is found for a given volatility.

Efficiency Frontier: In the interst of optimization, selecting the cases along the top edge reveals the stock tickers of interest. The slope of the top edge is where Sharpe ratio values are highest, (ref.2).





Cherry Criteria: We find the cases where Sharpe Ratio is above 2, volatility as less than 25%, price as less than $150, and beat percentage as above 10% in year 2018 on. From those cases, we accept the set of tickers who have accomplished this as our candidates. In this act of 'Cherry Picking', 81 tickers were found.

Indicator Performance Acceptance: In selecting stocks that seem to work well with this approach, we hope to diminish the downside. The indicator seems to complement the market in some cases, so we work with those over the cases where it didn't work out well.

Note the shape parameters chosen may be the deciding factor here – that other shape parameters such as the shape ordinal time span or composite vote scheme, might work better for other markets.

At this point, the indicator performance was deemed acceptable to progress to the next stage. However, we keep in mind that "*past performance is not necessarily indicative of future returns*."

Third Stage – Trade Classifier Development

Generating Trade data: "To error is human." So, imagine the stock trader isn't able to respond right away and the opportunity to buy or sell degrades. We want to rank trades by goodness of opportunity and classify good from not so good. Trade data is needed to do this kind of Supervised Learning.

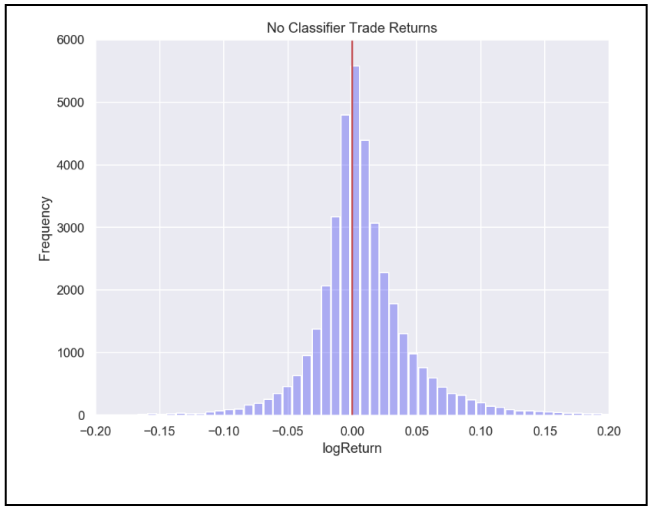
To generate this trade data, we simulate up to 5 days after HSC crosses thresholds (in the case it doesn't reverse direction in those five days). These could be considered sub-optimal trades because it is not what was intended by our trading rules. There are two sides to every trade, a buy first and then a sell in this long strategy. By having a balance of optimal with sub-optimal data, the classifier training improves.

Trade Mean Returns: In the case of a buy, the trade return may be any one of the 5 subsequent sell opportunities, so we report a mean return for that buy day against all those adjacent sells, and likewise

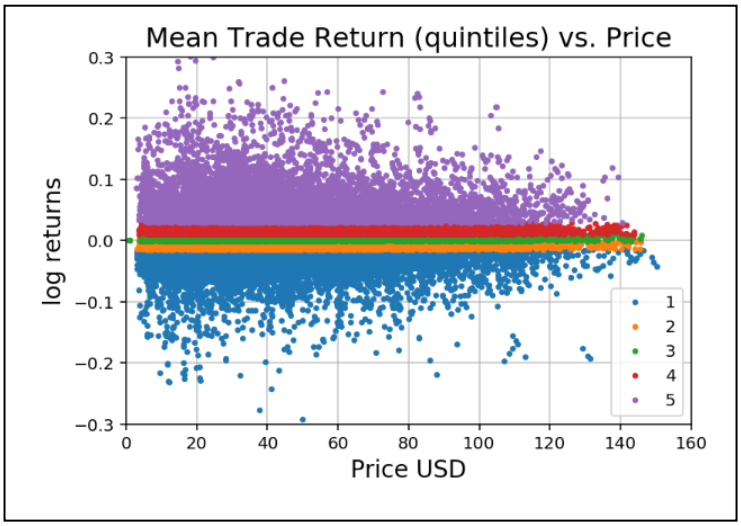
for the sell opportunities, there's five preceding buy days that might correspond with it so we'll return the mean value for that sell day. These mean return figures are a basic figure of merit by which a generated trade is judged. From these figures, a trade classification target is built.

ETL for Trade Data Generation: This is basically a spin-off of the earlier multiple stock survey ETL with additional trade masks that produce sub-optimal trade data. This culminates in a trade return graph that incorporates trade placement latency. This data goes on to train the classifier to decide good trade opportunities from not-so-good trade opportunities. The striving is to shift the odds for making more positive trades than not.

Mean Return Results: The distribution weighs heavier to the positive as seen by the +/-0.05 bars. The output data is a data frame of ~36,000 rows. There's a disproportionate number of buys and sells because in some cases the HSC reverses direction within the late period. Each row has either a buy or a sell trade and corresponding features such as date, ticker, estimated price, HSC and HSI components such as day shapes, shape ordinals, standard deviations, etc. It also has the target data, related to mean trade returns. These features go on to play a role in the trade classifier training.

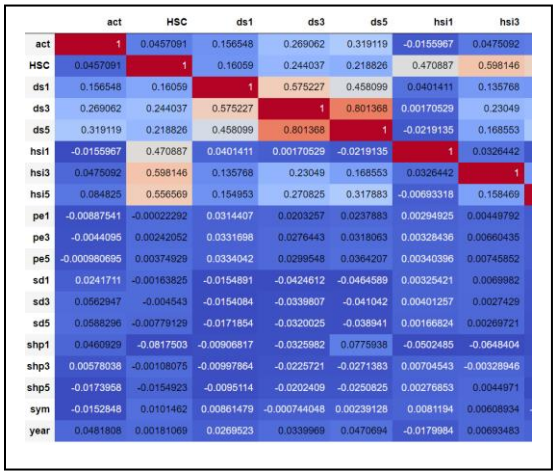


Trade Classifier Notebook: The ETL's generated trade data is imported into a classifier notebook called "trade_ranker.ipynb", this is where we train a classifier to predict a trade as good or not-so-good. This is an example of conventional Machine Learning classification, where a logistic regression model is chosen that affords both classification and probability ranking (ref.3.).



Trade Classifier Target: The mean returns are bin sorted into 5 bins of equal count called quintiles. The top two quintiles correspond to "High Return Trades." A target category called "Act" (short for Action) is based on this, where a "1" is assigned for trades in the top two bins and a "0" is assigned otherwise. The Act category is the target that we want the classifier to predict. Note that the middle quintile is above zero and acts as margin from negative trade returns.

Classifier Features: The features are used by the classifier to predict the target category. Features distilled in generating the Historical Shape Indicators are likely candidates here. These include the average price estimates (Open, High, Low, Close figures), the standard deviations, the day shapes, the shape ordinals, the HSI indicators, and the HSC. Furthermore, ticker symbols are rendered as categorical and 'year' is assigned as a feature to further this effort.



	act	HSC	ds1	ds3	ds5	hsi1	hsi3
act	1	0.0457091	0.156548	0.269062	0.319119	-0.0155967	0.0475092
HSC	0.0457091	1	0.16059	0.244037	0.218826	0.470887	0.598146
ds1	0.156548	0.16059	1	0.575227	0.458099	0.0401411	0.135768
ds3	0.269062	0.244037	0.575227	1	0.801368	0.00170529	0.23049
ds5	0.319119	0.218826	0.458099	0.801368	1	-0.0219135	0.168553
hsi1	-0.0155967	0.470887	0.0401411	0.00170529	-0.0219135	1	0.0326442
hsi3	0.0475092	0.598146	0.135768	0.23049	0.168553	0.0326442	1
hsi5	0.084825	0.556569	0.154953	0.270825	0.317883	-0.00693318	0.158469
pe1	-0.00887541	-0.00022292	0.0314407	0.0203257	0.0237883	0.00294925	0.00449792
pe3	-0.0044095	0.00242052	0.0331698	0.0276443	0.0318063	0.00328436	0.00660435
pe5	-0.000980695	0.00374929	0.0334042	0.0299548	0.0364207	0.00340396	0.00745852
sd1	0.0241711	-0.00163825	-0.0154891	-0.0424612	-0.0464589	0.00325421	0.0069982
sd3	0.0562947	-0.004543	-0.0154084	-0.0339807	-0.041042	0.00401257	0.0027429
sd5	0.0588296	-0.00779129	-0.0171854	-0.0320025	-0.038941	0.00166824	0.00269721
shp1	0.0460929	-0.0817503	-0.00906817	-0.0325982	0.0775938	-0.0502485	-0.0648404
shp3	0.00578038	-0.00108075	-0.00997864	-0.0225721	-0.0271383	0.00704543	-0.00328946
shp5	-0.0173958	-0.0154923	-0.0095114	-0.0202409	-0.0250825	0.00276853	0.0044971
sym	-0.0152848	0.0101462	0.00861479	-0.000744048	0.00239128	0.0081194	0.00608934
year	0.0481808	0.00181069	0.0269523	0.0339969	0.0470694	-0.0179984	0.00693483

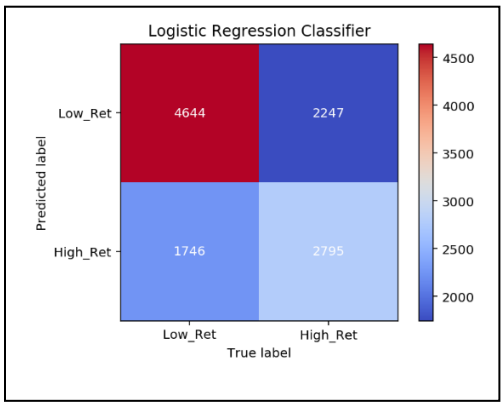
Feature Selection: All the feature candidates are studied in a correlation matrix. The aim is to include features that may correlate with the target but not with each other. In this case, the features seem weakly related. (Possibly there's room for more shapes that might improve the classifier training – because shape features were not found to be highly related.) Sci-Kit-Learns Recursive Feature Elimination (RFE) was applied to endorse these features as acceptable.

Test Train Split: The feature data was split as 70% for training and 30% for testing. However, the training data was then found to be disproportionate target class wise. To improve on this, Synthetic Minority Over Sampling Technique was applied by way of IMBLEARN's oversampling package called SMOTE. This equalized the training data for the two target class levels.

Model Implementation: Sci-Kit-Learns Logistic Regression model was implemented using an 'lbfgs' solver with multi_class 'multinomial' settings. The training for the solver resulted in a humble accuracy score of 0.65.

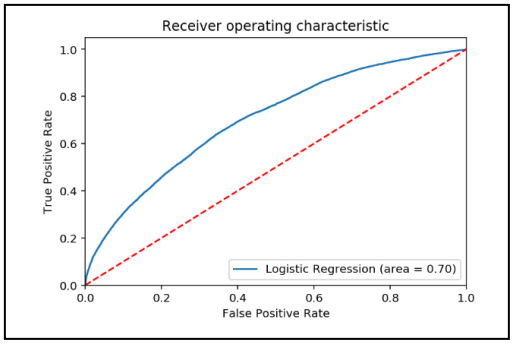
Model Classification Report: The precision and recall suggest an easier time precisely classifying the "Low Return" candidates.

	precision	recall	f1-score	support
Low_Ret	0.73	0.67	0.70	6891
High_Ret	0.55	0.62	0.58	4541
accuracy			0.65	11432
macro avg	0.64	0.64	0.64	11432
weighted avg	0.66	0.65	0.65	11432



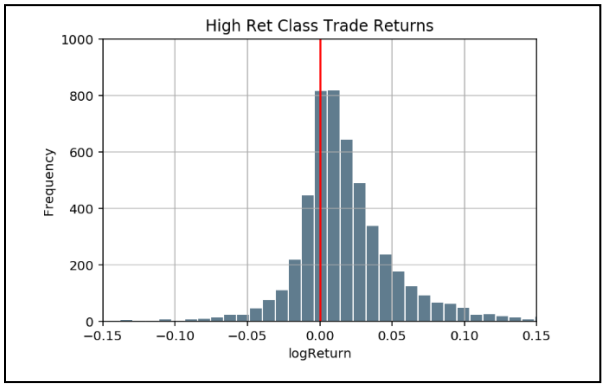
Model Confusion Matrix: This visually represents the classification report for the trained model's classification test performance.

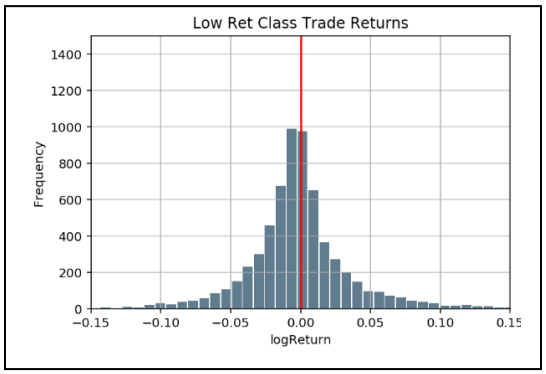
Model ROC Curve: The Logistic Regression behaves beautifully on the ROC graph.



Classifier Performance: Beyond typical classifier metrics, we now go back and apply the classifier to the test data's mean trade returns to consider if odds shift in favorable direction by use of the classifier. Note that the quintile bin sorting suggests an odds ratio of 2:3 in making a high return trade to not. The odds of a positive return are 1.37:1. As the trades are classified, exposure to opportunities is reduced as trade off to shifting this odds ratio.

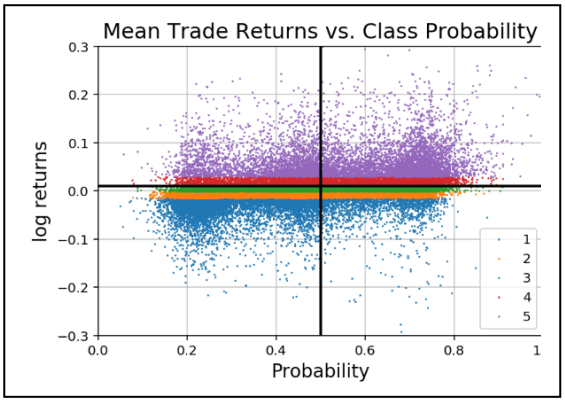
Mean Returns for High Return Class: The distribution is skewed positive as expected. The odds of a positive return are 2.92:1, the odds of a high return are 1.24:1. These are both up from the unclassified odds of 1.34:1 and 0.67:1 respectively and we want them to go up.

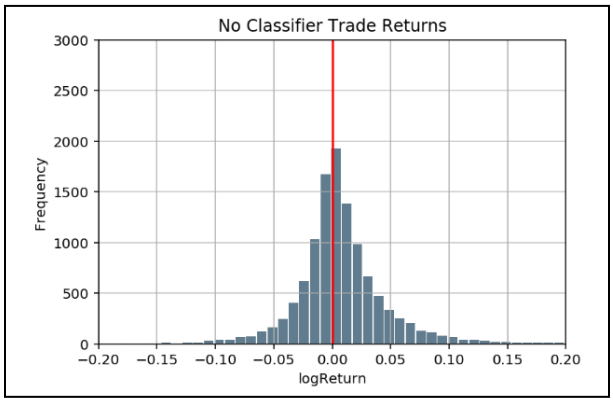




Mean Returns for Low Return Class: The odds of making a positive trade are 0.78:1, and the odds of making a high return trade are 0.38:1. These are both down from the unclassified odds as stated earlier and we want them to go down as we have no intention of trading them.

Trade Ranking: The logistic regression model not only predicts classes; it provides probability used in deciding that. As probability is a continuous function between 0 and 1, we simply sort trade opportunities by this prediction probability, where an opportunity with high opportunity is listed higher in advice than one of lower probability. It seems debatable how 'fair' this ranking scheme is, because of the spread in returns for any given probability.





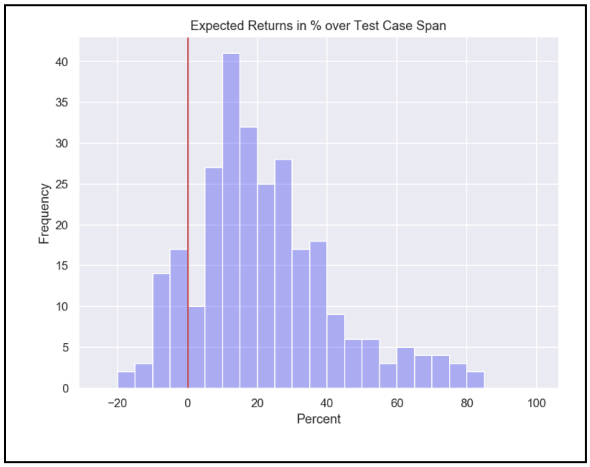
Accepting the Trade Classifier: Considering that the test data is a subset to the sub-optimal trade data, we can expect a punctual trader to have less latency and be less likely to fall into the diminished trade performance zone.

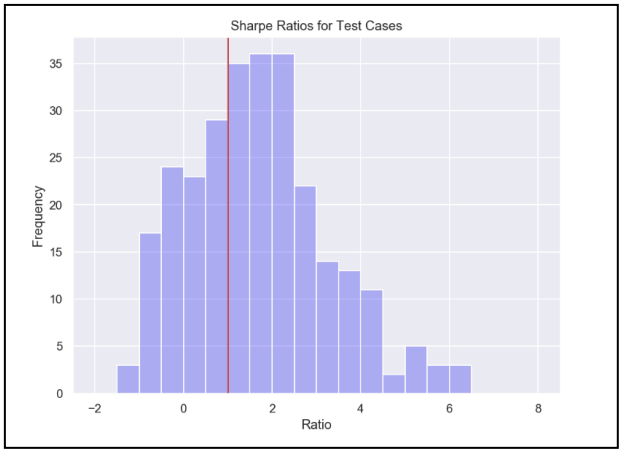
At this point the classifier performance is deemed preliminarily acceptable for proceeding to validation.

Adviser Validation ETL: here, all the pieces are put together to validate that the advised trades seem worthwhile. The data generated is similar to that built initially using "The Street" recommendations, however here we used selected stocks and apply the classifier to as a double check on top of the indicator's trade signals. For example, only when both the indicator and classifier agree, is a trade initiated. We expect that fewer trades will be made as a consequence, but this comes as an agreeable trade-off to having reduced exposure to the downside. The code to generate this data is "adviser_survey.ipynb".

Adviser Validation Analysis: Data from the ETL is imported and visualized in "adviser_analysis.ipynb." The span is the years 2017 to 2019, there's 276 test cases of 81 stocks, The mean price is $65, the mean number of trades per year is 5, the average days per trade is 29.

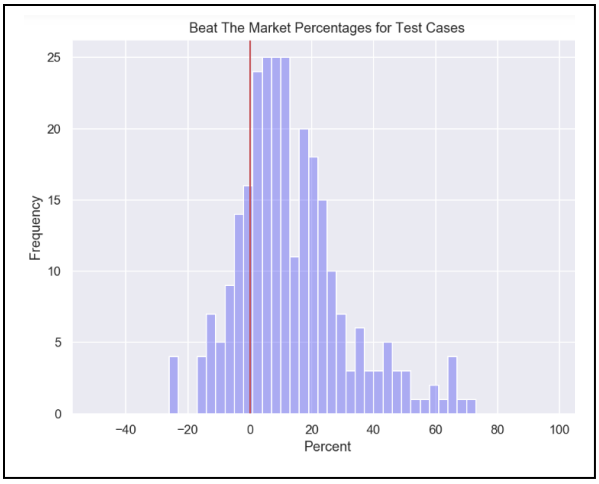
Validation Expected Returns: The expected returns span is one year in most cases. The mean value is 22% which is 4% better than before stock selection and classifier. The maximum drawdown was limited to -17% whereas it was complete ruin before. From the histogram, most trades advised seem positive.

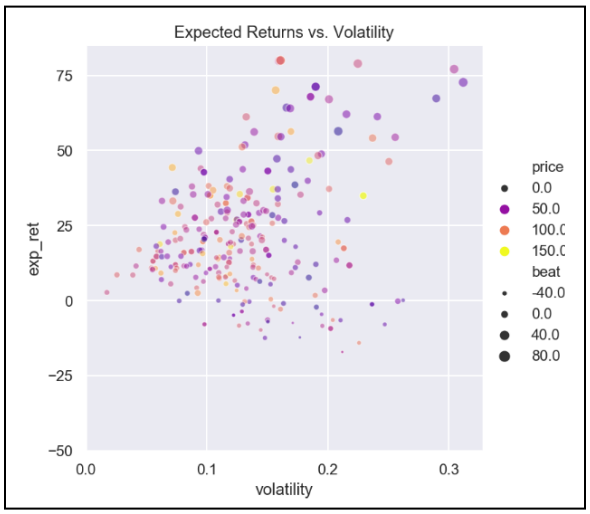




Validation Sharpe Ratio: The mean Sharpe ratio was found to be 1.7 which is good to very good performance wise. We wanted above 1 and the histogram peaks to the right side.

Validation Beat the Market Percent: The mean here is 13.6% for a predominantly positive histogram. The market beat percentages are the returns from the S&P500 index compared on similar 'in the market' trading days.





Validation Efficiency Frontier: There's that saying "Past performance is not indicative of future returns." Here we see it manifest where some fairly good past performers fell down below zero returns. However, the extent they fell seems relatively small.

Validation Acceptance: On the whole this approach exceeds expectation. The advised trades seem worthwhile. At this point we proceed to the final stage of the project.

Final Stage – Exercising Trained Models

Daily ETL: In this ETL, we start with selected stocks, apply the trained HSC features, compare the HSC features with their corresponding thresholds, distill trade opportunities, assign classification features, apply advising rules (top 25 per day), and generate a ranked table summary image for deployment.

Compact Data: Since the models are trained, we don't need to download "full" datasets. Here "compact" datasets of the last 100 days suffice. Actually, only the last 25 days are used to build ordinal shape representations for the current day. Most of the code blocks needed are reused as developed earlier.

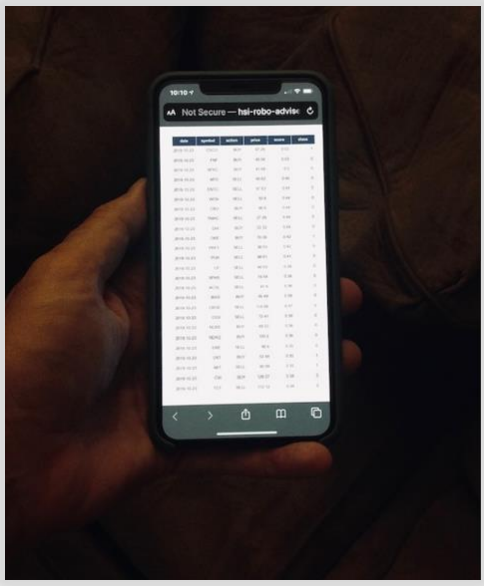
Implementation: The Jupyter notebook "daily_etl_102919.ipynb" and corresponding "etl.py" script captures steps needed for daily monitoring of the stocks of interest. Instead of running for a few hours as earlier, this ETL needs approximately a half hour to run. Unfortunately, data server errors need about a 10 second pause between tickers to fulfill queries without hiccups.

S3 Deployment: The daily ETL rank sorted summary table image is deployed on an AWS S3 static webserver using a boto3 based code snip, (ref.4) This is "S3_image_updater.ipynb" which requires credentials for S3 bucket access. The S3 static webserver is opened for public read access via policy. This is a minimally viable deployment for simplicity purposes. Independent upload is verified using S3 Browser tool (ref.5) The website is currently viewable at

http://hsi-robo-adviser.s3-website-us-west-2.amaxonaws.com

Phone Demo: The image is png format easily viewable on a mobile phone having web access. Notice the scores rank descending and the class of 1 or 0 (to Act or not) independently agrees in some cases.

date	symbol	action	price	score	class
2019-10-29	AUDC	SELL	20.5	0.75	1.0
2019-10-29	BRC	SELL	58.27	0.74	1.0
2019-10-29	MRK	BUY	84.43	0.73	1.0
2019-10-29	UPS	SELL	116.87	0.61	1.0
2019-10-29	NBHC	BUY	34.75	0.57	1.0
2019-10-28	LDOS	SELL	80.62	0.81	1.0
2019-10-28	WST	BUY	141.77	0.72	1.0
2019-10-28	GRMN	BUY	87.48	0.7	1.0
2019-10-28	UPS	BUY	116.4	0.65	1.0
2019-10-25	CCMP	BUY	149.79	0.91	1.0
2019-10-25	SAIA	BUY	101.34	0.85	1.0
2019-10-25	PRFT	BUY	38.68	0.76	1.0
2019-10-25	FSS	BUY	33.73	0.75	1.0
2019-10-25	RPM	BUY	70.78	0.73	1.0
2019-10-25	SPXC	BUY	42.5	0.71	1.0
2019-10-25	DRI	BUY	111.32	0.7	1.0
2019-10-25	CW	BUY	128.69	0.69	1.0
2019-10-25	ENTG	BUY	47.46	0.64	1.0
2019-10-25	ADI	SELL	106.13	0.58	1.0
2019-10-24	CCMP	SELL	148.56	0.86	1.0
2019-10-24	APO	BUY	41.34	0.73	1.0
2019-10-24	CW	SELL	127.73	0.71	1.0



The phone demo reveals some latency with S3 updates, the upload confirms the boto3 call, then ten to fifteen minutes later the updated picture (noticeable by date column) can be acquisitioned by phone.

Note that the image has dates sorted descending first, then score probabilities descending.

Requirements Review: In most cases, the requirements check. The classification accuracy in the 60% range came out strong because the target category affords margin. At this stage it's purpose as proof of concept demonstrates an innovative way to classify trade opportunities.

Another point of contention might be a disclaimer on the webpage – a comment to trade at your own risk; however, to understand the webpage you might need to read this report and then discover a disclaimer somewhere in this mix.

On the whole, the requirements review doesn't reveal glaring defects.

Areas of Future Improvement:

1) As mentioned early, the train/test split of 3-years prior to 1-year post is a generalization that seems to work in some cases but maybe not in others. Possibly, some markets do better with an HSI approach using a 5-year prior to 1-year post. It's an unexplored hyper parameter that requires compute resources to optimize. Note that it takes time for markets to make shapes and for the machine to learn them. In the case of an unknown shape, the last known shape value is adopted. With greater training spans, these shape gaps diminish.
2) The historical shape features chosen for this, 1-day, 3-day, 5-day with 7-day look-ahead were chosen off-the-cuff. These are other hyperparameter based features that could use optimization. Furthermore, the creation of better shape features would likely improve the

shape classification training. There's some degree of compute power needed, but this is par for the course with machine learning.

3) The stock universe is enormous and in this small study we ended up with 81 of tickers. Expanding to discover more opportunities is likely a fruitful adventure. Furthermore, markets such as cryptocurrencies likely have personalities with shape trainable features.
4) p-values that don't register significance likely indicate a market with some base trend that works with or without the shape indicator. Further investigating as to why the p-value significance varies, trend or no trend, may reveal insights leading to profit.
5) The project is not fully automated, it could likely play a part in an autonomous trade-bot.

Conclusion: In this project an adviser tool was built using machine learning practice to answer "What" and "When" for small time stock traders. A Historical Shape Indicator was developed and together with a high return Trade Classifier, this adviser is able to find and report reasonable trade opportunities.

References:

1) Top Rated Stocks ("The Street")
https://www.thestreet.com/stock-market-news/10579592/top-rated-stocks/top-rated-stocks.html
2) Efficiency Frontier
https://www.investopedia.com/terms/e/efficientfrontier.asp
3) Building a Logistic Regression in Python, Step by Step
https://towardsdatascience.com/building-a-logistic-regression-in-python-step-by-step-becd4d56c9c8
4) Export Pandas Data Frame as a Table Image
https://stackoverflow.com/questions/26678467/export-a-pandas-dataframe-as-a-table-image
5) S3 Browser
https://s3browser.com/

Disclaimer: this project is intended for educational purposes only and not recommended for real trading.