Intraday currency trend size prediction using Machine Learning

Zachary Vogel, Viraj Thakkar, Aajan Quail, Param Shah

Abstract - Foreign exchange (FX) markets facilitate the movement of trillions of dollars worth of capital around the globe, enabling a wide range of activities, including investment in emerging markets, hedging of foreign business risks, and international mergers and acquisitions. In this paper, we present a methodology to identify trends in data and provide a machine learning approach to identify whether or not the ongoing trend exceeds the median trend size. Several algorithms have been explored, and it is shown that the Adaptive Boosting (ADA) model proves to have the best performance, as indicated by its AUC (0.616) and accuracy (~60%). Although we have utilized much less data then it would be necessary for a model that could be deployed in markets, these initial results are quite promising, as marginal improvements in predictions will result in significant profits.

I. Business Understanding

FX desks are the backbone of the market's infrastructure – they enable the flow of capital across borders by trading currencies with customers. The desks trade currencies in pairs (Euro vs. the US Dollar – EURUSD, US Dollar vs. Mexican Peso – USDMXN) at prices that are dictated by market conditions. The price of a currency pair is listed as the amount of the second currency received in exchange for 1 unit of the first (1.10 EURUSD means you receive 1.10 USD for 1 EUR). A trader can make money simply by capitalizing on intraday changes in this price; for example, a trader might exchange USD for Euros at a price of 1.10 EURUSD, expecting the price to increase. If the price does in fact increase (let's say, to 1.21 EURUSD), the trader makes a 10% profit ((1.21-1.10)/1.10 = +10%). Conversely, if the price decreases to 1.00 EURUSD, the trader incurs a 10% loss. The trader's job thus boils down to predicting whether or not short term intraday trends will continue in the current direction, or reverse and go in the other direction.

When deciding on a trade, the trader must balance the potential costs and benefits of making the trade, given market conditions. This is no trivial task, as many factors influence currency prices at any given time. Current methods used to predict when to buy and sell currencies are fairly rudimentary in many cases, as they generally rely solely on the experience and intuition of industry experts. These methods are subject to a great degree of human error that can be easily mitigated with Machine Learning methods. Our project focuses on using a data-driven approach to solve this problem by employing Machine Learning to make general predictions about the size of intra-day currency pair trends. We motivate our task by first defining the following:

- A trend is an increase OR decrease in the price of a particular currency pair.
- The trend's size is the difference between the starting and final price of the trend.

The use of trends and trends sizes will be elaborated upon later in the paper, but we can essentially say that the goal of this project is to demonstrate that it is possible to predict whether or not a trend of meaningful size in a currency price, whether up or down, will exceed the median of all meaningful trend sizes across the entire data set. Here, the word "meaningful" is vague but is explained further in the data preparation section and is defined rigorously.

While predicting exact trend sizes or directionality of any market traded product is quite difficult and not feasible given the resources available for this project, it is our belief that our methods can lead to increased trader confidence by harvesting the relevant market data in real-time. Because of the enormous volume of currency traded on a daily basis, even marginal improvements in predictions could result in significant real-world value.

A key aspect to realize the underlying business function is that for any given transaction, the trader hypothetically has the option of hedging the risk with little to no loss, outside of opportunity cost. This means that we do not need to correctly predict and act on each trend instance, and can instead wait for the instances we feel we have some level of confidence/ information we feel brings our odds over 50%. In practice, this means only acting on instances where we believe our predictive accuracy is relatively high.

The data mining task at hand is to assemble a data set containing variables that are predictive of intraday price movements, synced up with the relevant currency price data, and then find some target metric that is predictable from this data in certain definable circumstances. As mentioned earlier, predicting the currency pair movement is not a trivial task due to the inherently unpredictable nature of currency markets. If these variables were clear/easy to identify, many market participants would already be utilizing them. Thus the most difficult part of the task is clearly the defining and processing of the variables based on subject matter knowledge. This subject matter expertise was available to us through a group member who works as an Emerging Market's currency trader for a bank.

II. Data Understanding

The data source for this project was real-time market data sourced from Bloomberg. Bloomberg provides 6 months of intra-day currency exchange rate data at 1-minute intervals. This volume of data suffices for our purposes. However, for full deployment, 2-3 years of data would be necessary, as this would allow maximal predictive power in the model. This amount of data is readily available for purchase and would be incredibly valuable for the purposes of a business.

After careful investigation, we decided it would be best to focus our efforts on a single currency pair. This greatly simplifies our model and minimizes the number of factors we need to take into consideration. We initially examined the movements of USDCAD (USD vs the Canadian Dollar), but eventually decided upon USDMXN (USD vs the Mexican Peso). An important part of our reasoning in choosing these currency pairs was due to their relative liquidity (resulting in reliable continuous data), relative stability (i.e. low likelihood of currency-specific events that have the potential to increase trend variances), correlation with other broad asset classes, and meaningfully large daily volatility (that is, large enough to consistently create enough meaningful intraday movement to assemble a large enough sample size). In the appendix, we have included the definition and logic for each variable chosen for prediction.

It is important to note that we purposefully introduced selection bias into our model by only using 6 months of currency exchange data. This was largely because it was free and easy to get, but beyond 2-3 years in the past, data would be less useful in generating predictions, given the continuous change in market dynamics and other important macro events that have occurred in the past few years (e.g. Trump/China trade war, Brexit, certain emerging market crises). In other words, even if we had access to 20 years worth of free data, we would have limited our dataset to at most 3 years, because data older than that would actually decrease our model's performance. Furthermore, we intentionally removed all rows with any missing data. While this slightly reduced the amount of data in our dataset, it further simplified our task. Our domain knowledge was simply not extensive enough to know how else to deal with these values - replacing nulls with the average of the column, for example, may not have been reasonable. Luckily, the null values were sparse enough to make a minor impact on the amount of data.

III. Data Preparation

The data preparation portion of the research was quite intensive, as a cohesive data set had to be constructed from scratch and analyzed through a series of data queries to Bloomberg's data archive. The first necessary decision was on the size of a "meaningful" directional move that was large enough to filter out noise, but small enough to capture a large enough sample set. Based on some trial and error, combined with subject-matter experience, we decided to define "meaningful" as ⅓ of the daily total expected intraday movement in the currency pair, based on the market price implied by options of 1Y annualized volatility for the currency pair. For USDMXN, this equated to a move of .2213%, based on 10.54 1Y Vol. With this, we also needed to decide how much of a retracement would define the end of a trend. Based on the most commonly used Fibonacci retracement level, we decided on the price returning to 68% of the maximum size of the trend [1].

The data wrangling process required importing minute by minute price data for the currency and relevant previous day/market open market variables, defining the price at the start of each day, and then running it through a function we defined which performs the following:

1. Checks the price at each minute vs the start of the day if no trends have yet occurred.
2. If the price difference is above the defined threshold (.22%), it declares the start of a trend, saves the price, time of this trend start, and the current maximum size of the trend.
3. For each minute after this while the trend is ongoing, checks if the overall price difference from the start of the trend to this minute is larger or smaller than the recorded max trend size. If it is larger the max size of the trend is updated. If it is smaller it checks if we have crossed the 68% of the max size threshold to end the trend.
4. If the trend ends, the trend starts information, synced up daily/ open variables, and max trend size is appended as an instance to the dataset we are assembling. The extreme price point which defined the max of this trend becomes the trend start point for the next trend we want to look at.

5. We then repeat this for each minute throughout the day, for each day in the dataset, comparing to either the relevant previous trend extreme or the start of the day.

Example: USDMXN starts the day at 19.00, 2 hours later the price crosses 19.043 and a trend is declared to have begun. It eventually trades up to 19.08 without moving meaningfully lower in the process, which is recorded as the trend extreme, and the percentage difference from day start (.00419%) is recorded as the absolute maximum trend size. The price then never goes above 19.08 before it crosses below 19.0544 ((0.08*.68)+19.00), at which point the trend is declared to have ended. 19.08 is referenced as the start of the next trend within the function.

Once the dataset of trends was assembled, we used the date-time from each trend start to pull in a series of real-time market data variables and join them to the constructed data set, which largely consisted of the value of technical indicators, index values, or index price changes on the day to that exact minute (this data was assembled and calculated manually separately).

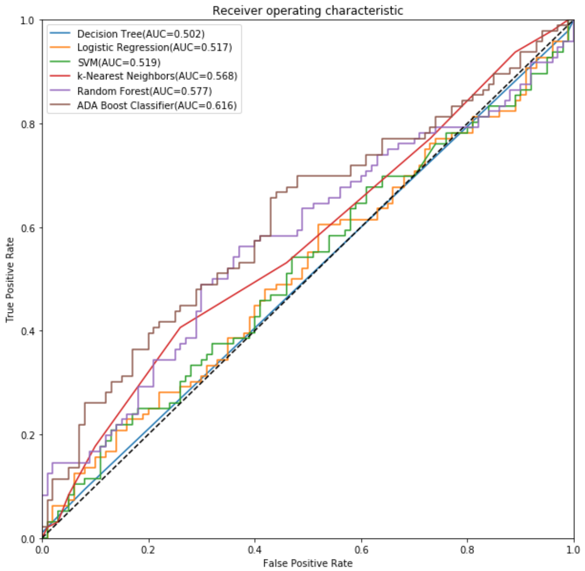
Finally, we needed to define a target variable we believed would be reasonably predictable based on the assembled variables. We initially tried working with the average absolute size of the maximum of a currency move within the training data, which amounted to trying to predict if the absolute percentage size of the move was above 0.35%. However, this resulted in very poor performance, and we realized that the average was being significantly distorted by large outlier moves that are quite typical to USDMXN. 0.35% was actually close to the 75th percentile of moves in the dataset; thus, any meaningful predictive results from this would likely imply leakage or an issue with the data itself. We finalized our target variable as to whether the absolute maximum trend size of each instance was above or below the median of the trends, or 0.29%, with a dummy variable of 1 for above and 0 for below, setting ourselves up for a classification approach.

IV. Modeling and Evaluation

After meticulous data wrangling and defining the target variable, our problem turned out to be a binary classification problem, allowing us to utilize a wide array of algorithms. We realized that the data needed to be split between training and testing based on time, as a random train-test-split resulted in overfitting that did not perform well in predicting future data (once again, we introduced selection bias into our training and test sets). Thus we used the first 70% of data in chronological order as training and the following 30% for testing. The initial approach used (base model) was a basic default decision tree algorithm, chosen primarily for its interpretability and insight to feature importances.

We decided to look at ROC & AUC as an evaluation metric for our decision tree because it provides insight into the range of performances across various thresholds and gives an easily interpretable score to compare models. Furthermore, returning an accurate prediction at a rate meaningfully above 0.50 at any given threshold would have significant real-world value; ROC/AUC, therefore, allows for flexibility in model finalization. The decision tree produced an AUC of 0.502. While the actual prediction accuracy of the decision tree was not high, observing the output of the decision tree graph allowed us to both interpret the feature importances used in the initial split, and apply subject matter expertise to sanity check if the algorithm's splitting criteria were reasonable or evidence of overfitting/data leakage was observed. While there did seem to be a connection between the lowest Gini nodes and positive performance, the sample sizes were too low for truly meaningful conclusions.

We subsequently tested Logistic Regression, KNN, SVM, and Random Forest models to varying degrees of success, essentially holding a bakeoff between them. In the end, we found an Adaptive Boosting(ADA) model to perform significantly better on a number of fronts. Below we included the AUC curves for the other models applied to the dataset, where ADA can clearly be seen to outperform (AUC = 0.616).

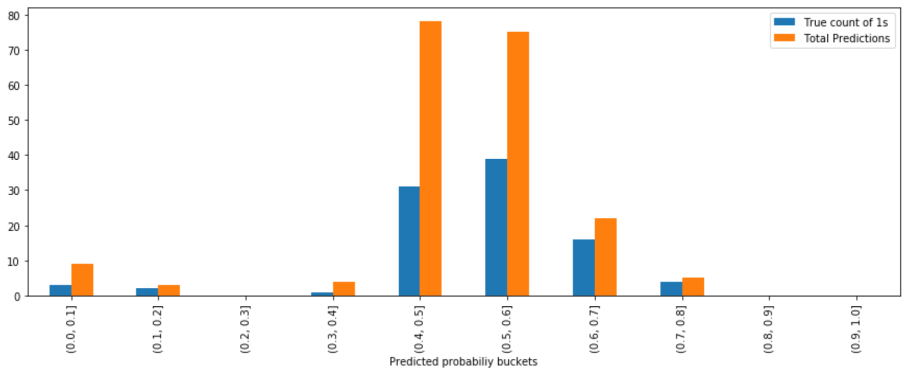


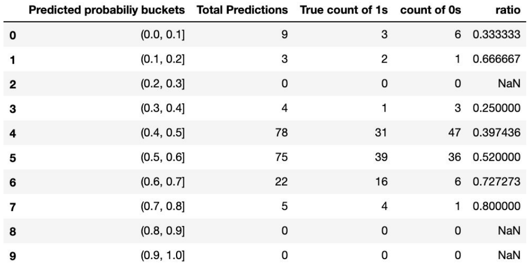
While a single decision tree uses a branching method to illustrate the possible outcomes, Boosted Decision trees (BDT) uses an ensemble of Decision Trees (“forests of decision trees”) which combines the output of all the trees. This process is called Boosting. Boosting is based on the idea of making a highly accurate prediction by combining the predictions of many “weak learners” (e.g. a small decision tree) [2].

Model Description for Adaptive Boosting (AdaBoost): There is an ensemble of small decision trees (max_depth < 5). Each tree attempts to correct the errors from the previous trees. The BDT algorithm combines forests of decision trees, each weighted according to their importance. The final output is the weighted sum of the output of all decision trees. The advantage of using the Adaboost model is that it gives us the prediction probability of the labels and is a good model for classification using a non-linear decision boundary. Also, the classifier is robust to outliers and almost optimum performance can be achieved with a few tweaks in the default parameters.

Nevertheless, we performed a grid search using cross-validation on the parameters of max_depth, learning rate and the number of trees in the ensemble and quickly realized that there would be a leakage from the future to the past due to the fact that cross-validation chooses at random the split for the validation set. In order to overcome this, we made use of a cross-validation object provided by scikit-learn called the ‘TimeSeriesSplit’ [3]. This allowed us to mitigate the problem of leakage during cross-validation making sure that we never use data from the future to judge each validation fold. Using this, we performed grid search and found the combination of max_depth=3, learning rate=0.01 and trees=700 to have the best performance in terms of the AUC metric.

The end goal of this modeling process is to maximize profit. In practice this means finding the decision threshold used on the prediction probability metric that returns the largest number of samples for which we are confident enough on the prediction to trade on, ideally resulting in positive profit & loss for a significant majority of attempts. To observe the optimal decision threshold we broke out the predicting probability scores along with the actual results into buckets of granular percentage ranges. Each bucket essentially represents precision, as it contains the number of test samples in that probability range (Total Predictions), as well as the number of 1’s that were actually present (True count of 1’s) using the AdaBoost classifier on the optimum parameters. Ideally, we would look to see less 1’s present in the buckets below the default threshold value of 0.5, and then want to observe if there is a linear relationship between further probability shifts from this threshold and the chances of 1's existing in the test set. For predicting smaller trends (label=0), we perform well overall (there are 60% true 0s in combined buckets below 0.5) and a linear trend in ratio seems possible, but there is not enough data outside of the 0.4 - 0.5 bucket to come to a definitive conclusion. However, for predicting larger trends (label=1) a linear trend in the ratio is quite evident, as we move from the precision of 52% for 0.5 - 0.6 to 74% for the combined 0.6-0.8 buckets. From a business perspective, this would mean a trader could draw real confidence that a predicted probability < 0.5 will actually result in a below-median trend and draw considerably more confidence that when the predicted probability > 0.5, a larger than the median trend is ongoing.





	Predicted probabiliy buckets	Total Predictions	True count of 1s	count of 0s	ratio
0	(0.0, 0.1]	9	3	6	0.333333
1	(0.1, 0.2]	3	2	1	0.666667
2	(0.2, 0.3]	0	0	0	NaN
3	(0.3, 0.4]	4	1	3	0.250000
4	(0.4, 0.5]	78	31	47	0.397436
5	(0.5, 0.6]	75	39	36	0.520000
6	(0.6, 0.7]	22	16	6	0.727273
7	(0.7, 0.8]	5	4	1	0.800000
8	(0.8, 0.9]	0	0	0	NaN
9	(0.9, 1.0]	0	0	0	NaN

V. Deployment

While we were able to accomplish positive results in our modeling approach to this problem, it is important to note that by no means would this be the end result of this project/ process in an actual business sense. As mentioned above, we limited ourselves to only 6 months of data and did not extensively explore potential features. The data available in this field for anyone interested in pursuing a similar project is incredibly abundant and rich; thus, with minimal time, money, and human capital restraints, we believe that an extremely powerful model could be built. However, the amount of investigative work required for each potential variable in the model is quite time-intensive, and our group was simply unable to build anything more complex than what is presented in this paper given the restraints. The true end goal of this project is simply to provide proof of concept on the process and demonstrate general predictability of the size of intra-day currency movements. As it stands, our model is not ready for deployment.

However, assuming the results are confirmed on a larger dataset/continues to reach a point that gives us enough confidence that we feel it significantly augments the current decision-making process, deployment would be fairly straight forward. A data feed of the relevant real-time price and variables from Bloomberg would be continuously pulled throughout the day and filtered into relevant trends as they start and end. When we had the need to predict the results of an ongoing real-time move we would take the relevant data for the start of the trend, and feed it into our trained model. The trader would already have a market view one way or another as they assess whatever risk position is in question, and would then take into account the results of the model. If the model agreed with their assessment one way or another they would be given an additional bit of confidence in their decision, perhaps allowing them to have a bit more tolerance for short term volatility in a negative direction with a belief the eventual price trend will end up positive for them, or hold a larger portion of the risk longer without early hedging, which would further maximize profit on a winning trade. Vice versa if the model disagreed with their own assessment, it would cause them to pause and reassess, at worst serving as a sanity check, while maybe causing them to rethink and change their end decision.

The continuous evaluation of the model is fairly straightforward as well, as one would periodically (perhaps monthly) add the new data produced each day to the training data set. One would also need periodically retest the entire data set on various total time lengths vs predicting on test sets of recent data to ensure that training data produced increasingly long ago remains predictive, as market conditions change continuously and can undergo major shifts that would compromise our results.

There are crucial things a firm would have to recognize when deploying this process. The underlying prediction is binary and does not predict anything currently about the course or size of a trend once it exits the narrow confines of the model. The model depends heavily on human judgment in interpreting the findings, and blindly following the model without considering the individual market conditions of the current trend in question could result in significant losses. Anyone deploying the results of the model would have to understand the limitations and context of the probabilities output by the process, as trusting too fully in or misunderstanding the results could also result in the decision-making process worsening rather than improving, causing added risk to high leverage situations.

As noted earlier, the current experiment was done on a limited data set and trusting large sums of bank capital to a decision-making process underpinned by such a limited data set would be risky. The model would only be augmenting current decision-making processes for ongoing business in markets that are quite robust, meaning our model's deployment at a single bank would have little to no effect on wider market conditions. In theory, if a majority of large banks all deployed similar processes looking at the same data and variables, dangerous levels of groupthink and herding into the same investments could occur, which could cause large losses and market impact when market conditions change. This scenario is highly unlikely though. The transactions in question for intended model usage are between educated market professionals and do not affect ordinary consumers/ investors. Thus the only major ethical concerns we can envision are the possibility of a trader aided by model results engaging in dangerous levels or proprietary trading beyond job description/ risk limits. This can be contained by risk management processes already in place, but risk management personnel should clearly be looped into whatever metrics traders are using for their decision-making process.

VI. References

[1] Investopedia.com: Fibonacci retracement level

[2] Yoav Freund, Robert E. Schapire. 1999. "A Short Introduction to Boosting." *Journal of Japanese Society for Artificial Intelligence* 771-780.

[3] Sklearn documentation: TimeSeriesSplit

[4] Forex Technical Indicators

[5] Bloomberg

VII. Github Code Link

https://github.com/Param9498/DS-GA-1001-Project

VIII. Appendix

Feature Description

- Trend Start Time -Time when the % size of the directional currency move we are currently observing crosses the trend threshold we established. This is the moment when all of the feature variables used in the prediction model are taken as snapshots. To be clear, the trend has already been ongoing before this time, but this is the time it has grown large enough that we now deem it to not be noise and add it to our trend data set.
- 1M Vol Open – The opening price of the day for 1-month volatility of the underlying currency pair. This equates to the rate that, if annualized over a year, the currency pair is expected to move of the next month.
- 1Y Vol Open – The opening price of the day for 1-year volatility. This is an annualized rate already, so this equates in percent to how much we would expect the currency to move over the next year. All else being equal if you bought an options contract and stripped out the other underlying risks, if the currency moves more than this percent in the next year, you would make money. If it moved less you would lose money.
- DXY Open – The opening price of the day for the DXY Index, which is a weighted basket of currencies vs the US dollar. If this value is increasing it means the USD is getting stronger on average vs the basket of currencies.
- Daily Trend - Which sequential number trend of that day the current trend is. So for example, if this variable equals 3, this is the 3rd observed trend large enough to be added to our dataset this day.
- Previous Day DXY Trend – This is, as of yesterday's close, the % change in the DXY index over the previous 5 days. This variable is meant to indicate whether the USD is currently in the midst of a strong medium-term trend one way or another.
- Previous Day RR – Yesterday's "Risk Reversal" price, which is implied volatility of call options minus implied volatility of put options for the underlying currency and tenor (overnight in this case). This reflects the market's expectations for the likelihood of an outsized move either higher (calls more expensive) or lower (puts more expensive).
- Previous Day RR MAVG – The 10-day moving average price for Risk Reversals. Here so we can see if the current RR reflects a meaningful shift from previous expectations in the medium term.
- Previous Day SPX Trend – Previous 5-day trend in the S&P500, reflects a medium-term trend of US stocks.
- Trend Start – The price of the underlying currency at the moment we register the trend, i.e. it crosses the threshold to be considered a trend.
- Trend Type – Whether this is an uptrend in the currency pair (USD gaining value up, currency 2 losing value), or downtrend.
- USDXXXV1M Tick – The value of 1-month volatility in the currency pair at trend start time. Used to determine the change on the day to the point of the trend starting of 1-month volatility.
- USDXXXV1Y Tick – Same as 1 month, but for 1-year volatility.
- SPX Tick – Value of S&P 500 at trend start time.
- DXY Tick – Value of DXY index at trend start time.
- V1M Change – The raw change in the 1-month volatility from the days open to trend start time.
- V1Y Change – Same as 1M, but for 1 year.
- DXY Change – Raw change in the DXY index from open to trend start time.
- SPX Change – Raw change in the S&P index from open to trend start time.

- MACD - Moving Average Convergence Divergence (MACD) is a trend-following momentum indicator that shows the relationship between two moving averages of a security's price. The MACD is calculated by subtracting the 26-period Exponential Moving Average (EMA) from the 12-period EMA.
- ROC 4 – Rate of change from 4 periods earlier.
- ROC 10 – Rate of change from 10 periods earlier.
- MXEF – An index tracking a basket of emerging market stocks.
- MXEF CCY – An index tracking a basket of emerging market currencies.
- BBDXY – Bloomberg Dollar Index, a currency index similar to DXY but with more even weightings.
- RSI - The relative strength index (RSI) is a momentum indicator that measures the magnitude of recent price changes to evaluate overbought or oversold conditions in the price of a stock or other asset. It is displayed as an oscillator (a line graph between two extremes) and can have a reading from 0 to 100.
- Target – Computed by checking whether the "Absolute Max Trend Size" of the trend instance in question is above or below average. The average is computed across all of the instances in the data set. If the value is 1 it is above average, 0 if below.

Contributions

Zach Vogel (zjv207) -

- Currency trader with subject matter expertise in this area so defined project goals and provided guidance into the direction of approach.
- Performed data pulls/wrangling and decided upon the majority of variables using expertise.
- Guided content in the majority of paper sections due to the ability to cohesively speak about the real-world goals of the project.

Viraj Thakkar (vt943) -

- Implemented the Boosted Decision Trees (Adaboost) model which eventually gave the best performance and helped in the insights/concepts of thresholds and AUC.
- Checked that default threshold value of ~0.5 gives optimum performance in terms of testing accuracy and precision (i.e. high true positives, which is what we need).
- Briefly explained AdaBoost and its advantages over other models. Provided interpretation in probability buckets bar graph and helped in a grid search of parameters.
- Edited, over multiple iterations, the model & evaluation section while collaborating with Param.

Aajan Quail (aqd215) -

- Implemented Logistic Regression and SVM models. These provided only slight improvements upon the baseline model. Assessed model performance using a confusion matrix and performed grid-search over regularization parameters.
- Contributed to the probability buckets bar graph and its interpretation. Helped in grid search of parameters.
- Finalized writing of all sections by providing a clear, cohesive explanation for each section, and served as the main editor for the final paper.

Param Shah (prs392) -

- Implemented Decision Tree and Random Forest models. Provided intuition on feature importance, target-class imbalance and train-test split bias.
- Integrated everyone's code and built all the plots (AUC and Bar plots)
- Researched and implemented time-series cross-validation for model selection on all models.
- Improved efficiency of the trend-detection algorithm (from 15 mins to 30 seconds for 6 months of data).
- Edited, over multiple iterations, the model & evaluation section while collaborating with Viraj.