How to train a binary classifier (AI) to detect options fuckery: By u/broccaaa

How to train a binary classifier (AI) to detect options fuckery

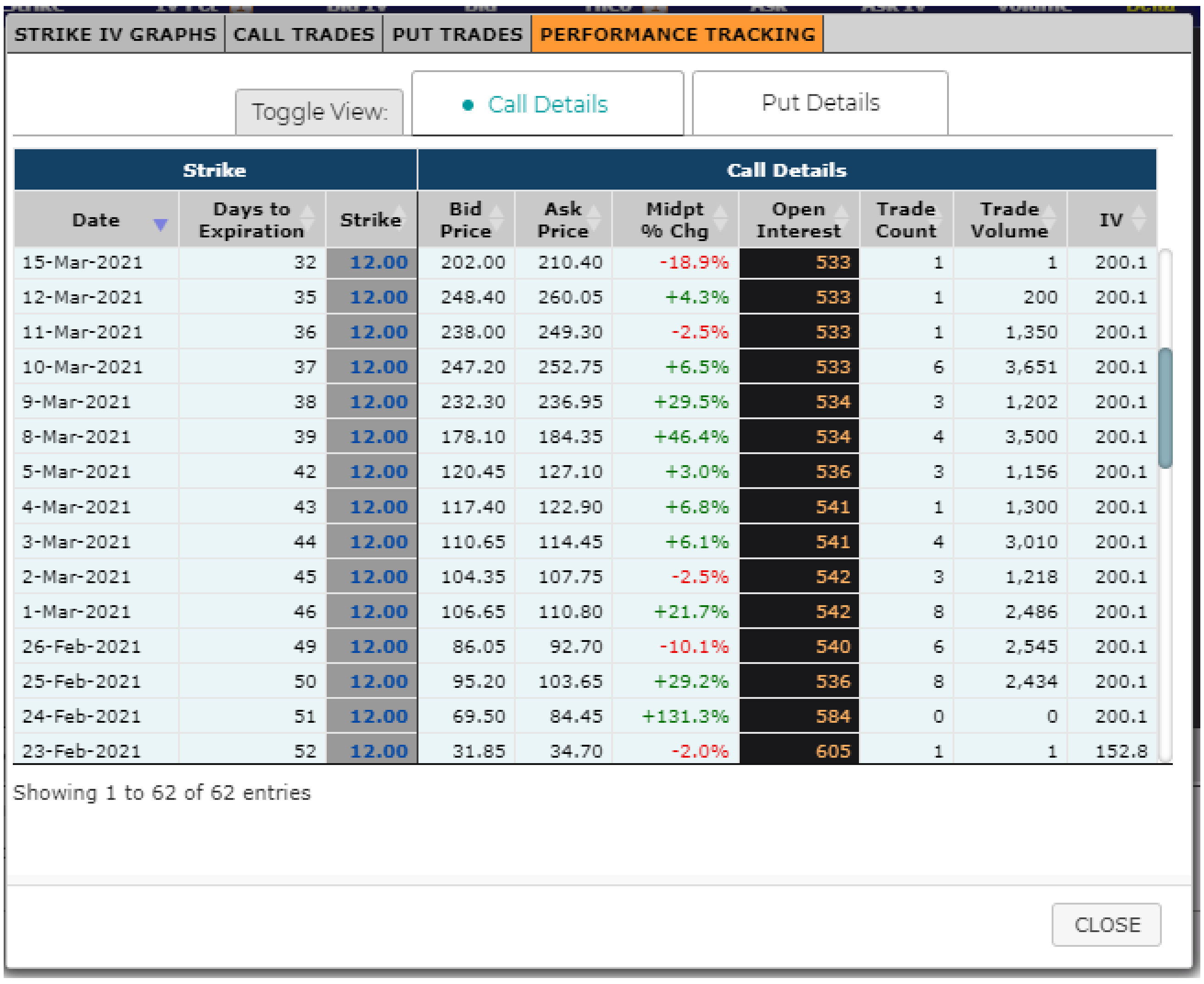
.

Discussion □

There were a lot of comments requesting more details on the methods I used to train the AI discussed in my recent DD post . Here I will provide all the key details and also discuss some of the potential challenges and biases with the approach. I'll try to make it as approachable as possible for anyone interested.

The problem

One way that a naked short seller can 'resolve' their FTDs without actually covering is through options fuckery. Deep in-the-money (ITM) calls can be bought and exercised immediately to acquire the shares and close the FTDs. The SEC published a paper on this ILLEAGAL practice. Read this paper if we really want to understand how these tricks are used by shorts.
Other great DD has been posted showing when Deep ITM volumes have been used to cover FTDs.
I will focus on some of the work by u/dejf2 but a number of other apes have been checking and reporting back on weird deep call volumes. Here is an example from u/dejf2:



Strike			Call Details						
Date	Days to Expiration	Strike	Bid Price	Ask Price	Midpt % Chg	Open Interest	Trade Count	Trade Volume	IV
15-Mar-2021	32	12.00	202.00	210.40	-18.9%	533	1	1	200.1
12-Mar-2021	35	12.00	248.40	260.05	+4.3%	533	1	200	200.1
11-Mar-2021	36	12.00	238.00	249.30	-2.5%	533	1	1,350	200.1
10-Mar-2021	37	12.00	247.20	252.75	+6.5%	533	6	3,651	200.1
9-Mar-2021	38	12.00	232.30	236.95	+29.5%	534	3	1,202	200.1
8-Mar-2021	39	12.00	178.10	184.35	+46.4%	534	4	3,500	200.1
5-Mar-2021	42	12.00	120.45	127.10	+3.0%	536	3	1,156	200.1
4-Mar-2021	43	12.00	117.40	122.90	+6.8%	541	1	1,300	200.1
3-Mar-2021	44	12.00	110.65	114.45	+6.1%	541	4	3,010	200.1
2-Mar-2021	45	12.00	104.35	107.75	-2.5%	542	3	1,218	200.1
1-Mar-2021	46	12.00	106.65	110.80	+21.7%	542	8	2,486	200.1
26-Feb-2021	49	12.00	86.05	92.70	-10.1%	540	6	2,545	200.1
25-Feb-2021	50	12.00	95.20	103.65	+29.2%	536	8	2,434	200.1
24-Feb-2021	51	12.00	69.50	84.45	+131.3%	584	0	0	200.1
23-Feb-2021	52	12.00	31.85	34.70	-2.0%	605	1	1	152.8

Showing 1 to 62 of 62 entries

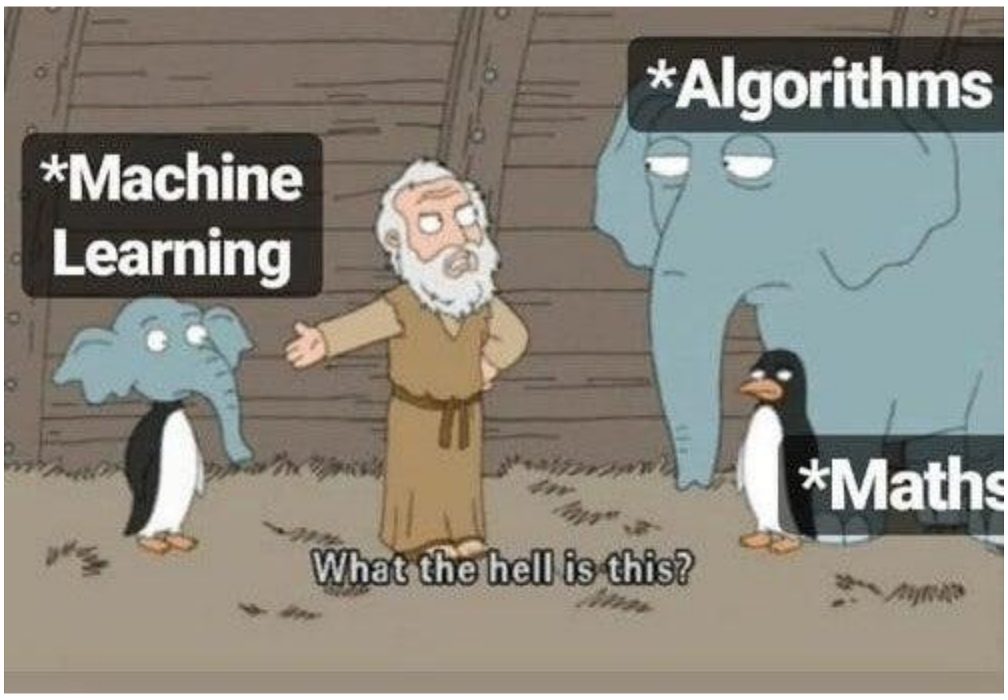
APR16 12C - 24052 contracts traded while Open Interest changes by 3 - between FEB 25 and MAR 12, or 2.405m FTDs reset.

This is one hallmark of fucky options trading. Volumes way way larger than open interest and very little increase in open interest.
BuT bRoCCaAa tHeSE cOulD bE fRom noRmAL tRAdInG oR arBItraGe!

Do you really think the same 500 contracts changed hands an average of 48 times in just over a week when there was zero volume on previous days? And then if new contracts were bought can you explain why not a single one of these contracts was held longer than a day?
Neither of these explanations make sense of this. The activity in this example was on strike prices that are more than 90% lower than current GME price.
Also look at Mar 11. 1 trade made of 1350 contracts with an open interest of only 533 and not a single increase in open contracts the following day. The same is seen on March 4th.
Take a look at more of u/dejf2's posts to understand his findings and how he identified deep call fuckery.

What is an AI and why do I need it?

When people talk about AI they are really talking about machine learning. It doesn't matter if they're Google translating all the world languages. Or DeepMind (also owned by Google) training a machine to learn how to play chess or GO and beat the best players in the world. All of these methods come under the umbrella of machine learning. So what is it?



Machine learning combines statistical modelling and computer science advances into a framework that allows algorithms to learn from past events (often labelled data) and then predict future observations.

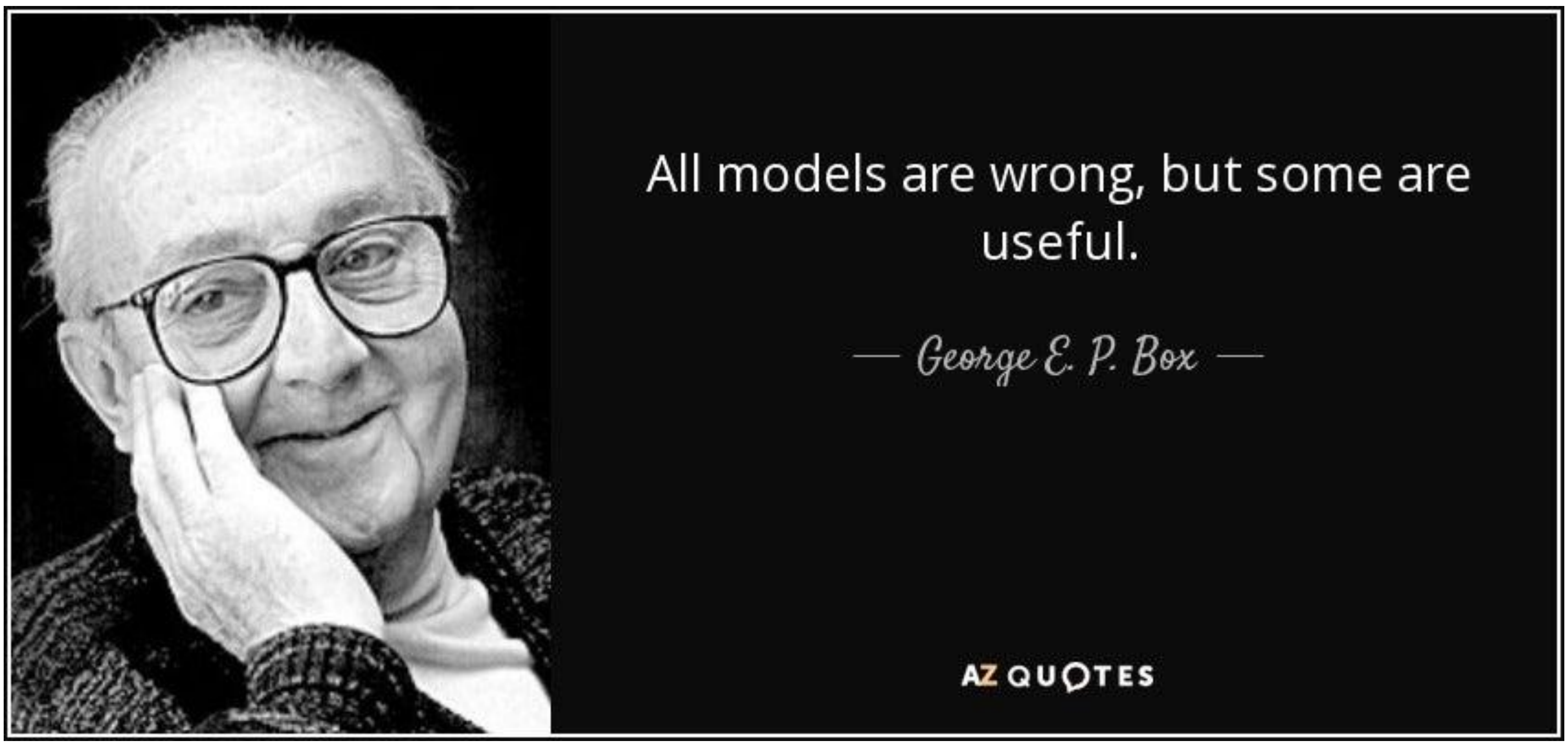
Many type of machine learning algorithms and approaches exist. A common distinction is between classification versus regression. Identifying cases of Deep ITM options fuckery falls under the classification set of problems.

The main reasons for wanting to build a classifier rather than label everything myself are as follows:

1. Once trained the classifier can label further data instantly
2. Given a good enough training dataset future labelling will not have the problem of human error when distracted or going through thousands of rows of data
3. The ability of the classifier to train and then be tested on unseen data provides a way to sanity check the labelling scheme used - if an inconsistent labelling was performed then model performance will be poor.

A method for classifiying Deep ITM call fuckery

I want to state this very clearly at the start. The goal when building this model was not to create the bestest most amazing of all classifiers ever. I wanted something useful. Something reliable. Something interpretable. And also something quick because I'm not spending all my time squeezing an extra 1-2% accuracy out of a model that is already useful.



Here are the main steps required to train a classifier:

1. **Get data**: Gather historical options data with as much granularity as possible (I got end of day data for all strike and expiry's)
2. **Select training data**: Select a representative subset of the data to be used for training the model
3. **Hand label the selected data**: 1 for suspected fuckery; 0 for normal or uncertain trading.
4. **Examine biases**: Once labelled it is important to check for things like imbalances in the dataset (e.g. fewer fuckery examples to normal examples).
5. **Develop features**: This is the secret sauce. After all the experience you gained while labelling the data try to create features that capture the key criteria used to make the label. Make sure you develop other features that could model other potential biases (e.g. strike-price to share-price ratio).
6. **Split data**: Divide the data into training and test sets (I am not using a validation set because *I can't be fucked with optimising all hyperparameters*). Use methods that can account for biases if needed.
7. **Train models**: Use the training dataset and the features you created to train the model. This has multiple steps which I'll detail below. One thing to remember is to normalise your data before training. Multiple models can tested to see which one works best. Also a good idea to do feature selection to make sure all the information you give the model is helpful.
8. **Test models**: Apply the trained models to the separated test data. *It is absolutely critical that the test data was never previously seen by the algorithm* - avoid data leakage.
9. **Model selection**: If multiple models were tested (different algorithms or hyperparameters) use an unbiased score to select the best ones. Many different scores exist but F1-score is a good place to start.
10. **Model prediction**: If you are happy that you have a classifier that can predict data almost as well as you could yourself (high accuracy score, few biases etc.) then apply the model to your whole dataset, including all the data you didn't label at the start. Make sure the *exact same feature development and normalisation steps are applied* to the new data before prediction.
11. **Explore the results**: You now have labels for all your data thanks to the initial effort you put into labelling the training data and then training the classifier. If you did it well the classifier might label data as well as you (or even potentially better). Plot your results and interpret the findings.

My methods

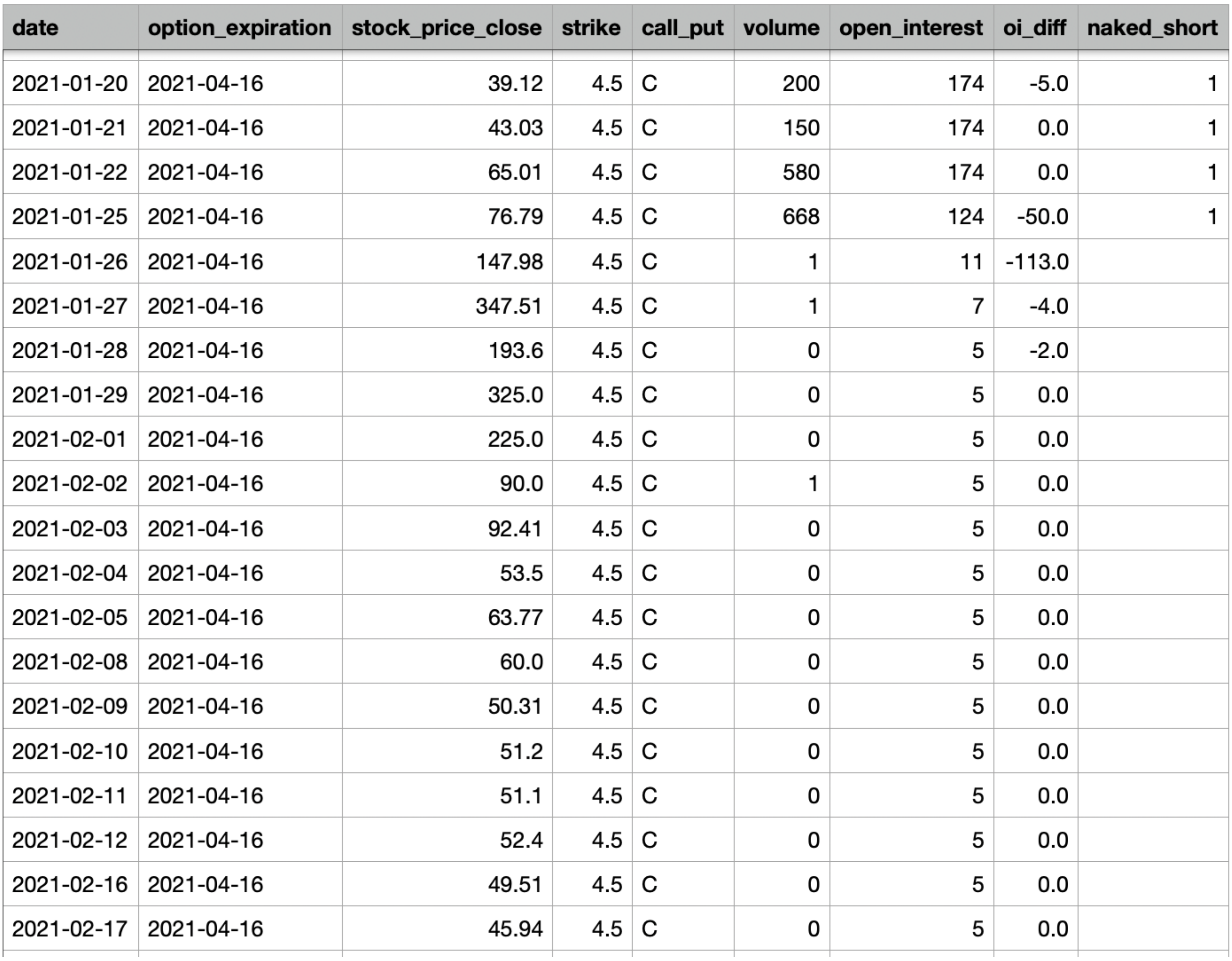
Get data: Historical options data was gathered from an online data supplier. Market chameleon is a good place with a free trial. I paid another service for over a year's worth of GME options data.

Select training data: I selected all data from Jan 20th - Feb 20th. This includes the Jan mini-squeeze and a lot of the days with known fuckery based on other DD. I only included data where strike price was less than 70% of the share price. This might miss some fuckery but I preferred to be conservative here. I had a total of 10204 rows of data in my training data.

Hand label the selected data: I followed a labelling scheme similar to what was done in past DD's and the examples I have above.

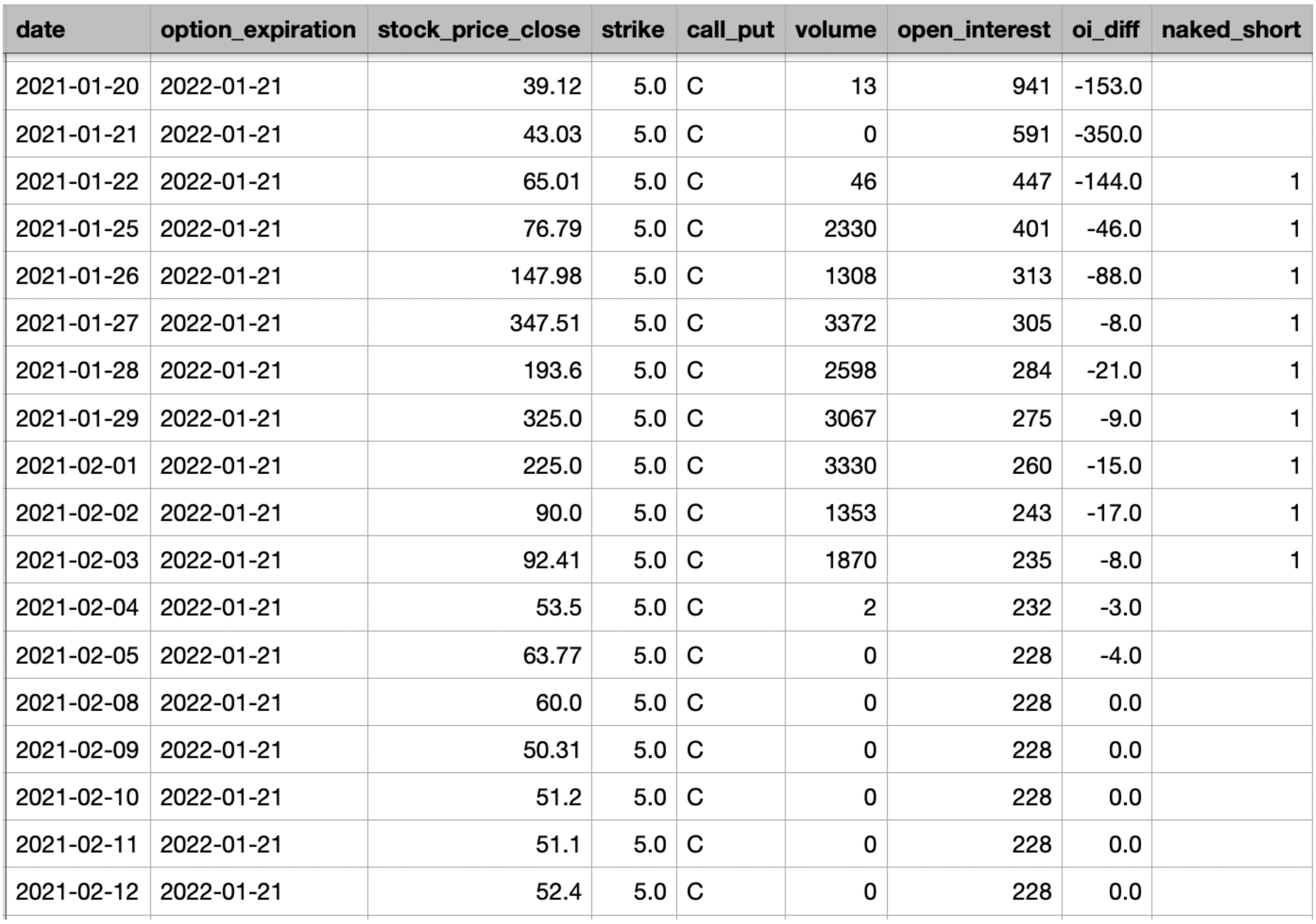
Note: *This is the key step where we might be getting biases. The model can only perform as well as the labeled data I give it!!*

Here are some examples of data I chose to label as suspicious or normal/unknown:



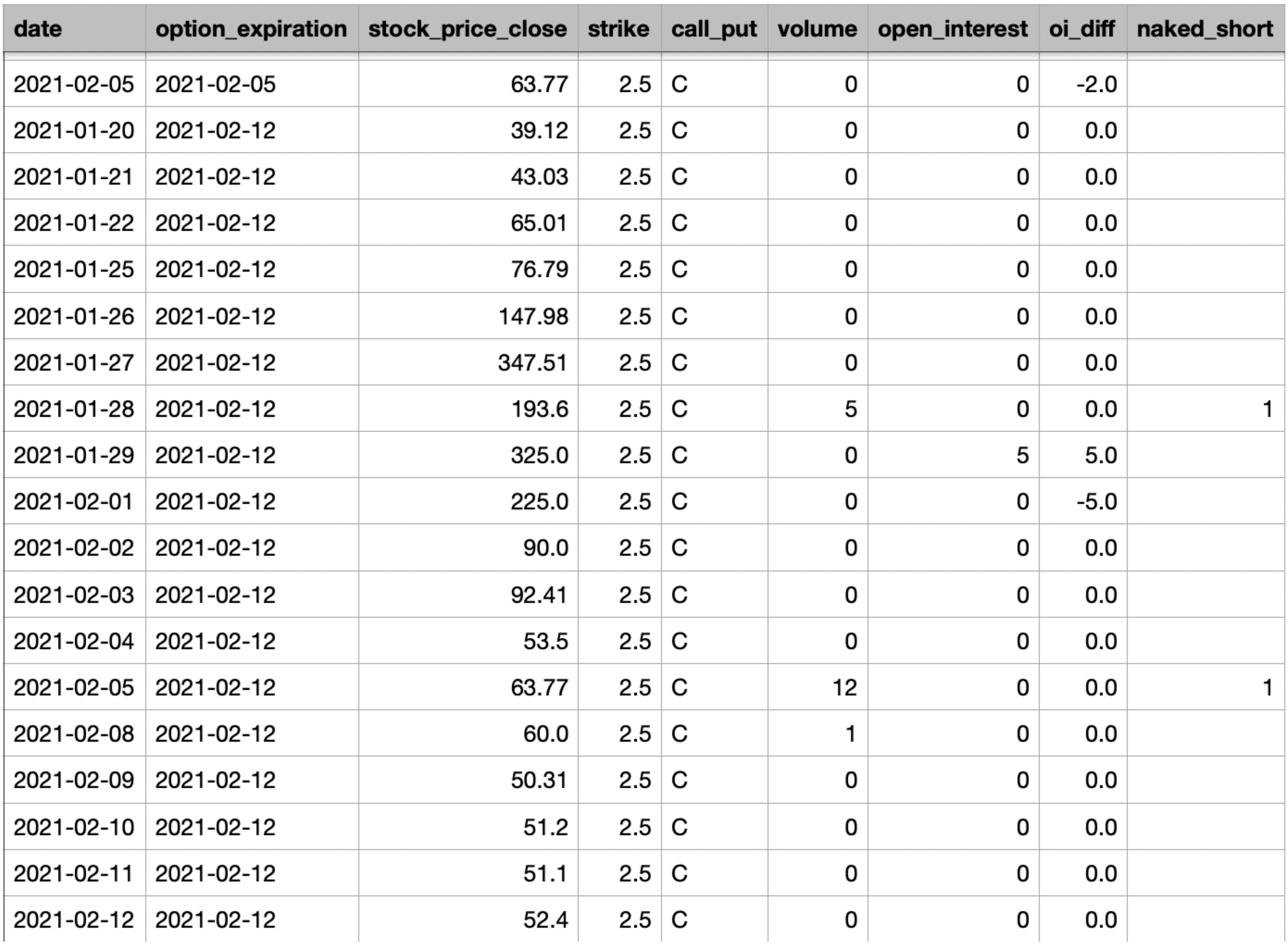
date	option_expiration	stock_price_close	strike	call_put	volume	open_interest	oi_diff	naked_short
2021-01-20	2021-04-16	39.12	4.5	C	200	174	-5.0	1
2021-01-21	2021-04-16	43.03	4.5	C	150	174	0.0	1
2021-01-22	2021-04-16	65.01	4.5	C	580	174	0.0	1
2021-01-25	2021-04-16	76.79	4.5	C	668	124	-50.0	1
2021-01-26	2021-04-16	147.98	4.5	C	1	11	-113.0	
2021-01-27	2021-04-16	347.51	4.5	C	1	7	-4.0	
2021-01-28	2021-04-16	193.6	4.5	C	0	5	-2.0	
2021-01-29	2021-04-16	325.0	4.5	C	0	5	0.0	
2021-02-01	2021-04-16	225.0	4.5	C	0	5	0.0	
2021-02-02	2021-04-16	90.0	4.5	C	1	5	0.0	
2021-02-03	2021-04-16	92.41	4.5	C	0	5	0.0	
2021-02-04	2021-04-16	53.5	4.5	C	0	5	0.0	
2021-02-05	2021-04-16	63.77	4.5	C	0	5	0.0	
2021-02-08	2021-04-16	60.0	4.5	C	0	5	0.0	
2021-02-09	2021-04-16	50.31	4.5	C	0	5	0.0	
2021-02-10	2021-04-16	51.2	4.5	C	0	5	0.0	
2021-02-11	2021-04-16	51.1	4.5	C	0	5	0.0	
2021-02-12	2021-04-16	52.4	4.5	C	0	5	0.0	
2021-02-16	2021-04-16	49.51	4.5	C	0	5	0.0	
2021-02-17	2021-04-16	45.94	4.5	C	0	5	0.0	

Volume similar to or larger than open interest for multiple day in a row with no change in open interest. After this activity all open contracts are exercised and open interest drops to just 5 contracts. No one is interested in any trading after the suspicious window.



date	option_expiration	stock_price_close	strike	call_put	volume	open_interest	oi_diff	naked_short
2021-01-20	2022-01-21	39.12	5.0	C	13	941	-153.0	
2021-01-21	2022-01-21	43.03	5.0	C	0	591	-350.0	
2021-01-22	2022-01-21	65.01	5.0	C	46	447	-144.0	1
2021-01-25	2022-01-21	76.79	5.0	C	2330	401	-46.0	1
2021-01-26	2022-01-21	147.98	5.0	C	1308	313	-88.0	1
2021-01-27	2022-01-21	347.51	5.0	C	3372	305	-8.0	1
2021-01-28	2022-01-21	193.6	5.0	C	2598	284	-21.0	1
2021-01-29	2022-01-21	325.0	5.0	C	3067	275	-9.0	1
2021-02-01	2022-01-21	225.0	5.0	C	3330	260	-15.0	1
2021-02-02	2022-01-21	90.0	5.0	C	1353	243	-17.0	1
2021-02-03	2022-01-21	92.41	5.0	C	1870	235	-8.0	1
2021-02-04	2022-01-21	53.5	5.0	C	2	232	-3.0	
2021-02-05	2022-01-21	63.77	5.0	C	0	228	-4.0	
2021-02-08	2022-01-21	60.0	5.0	C	0	228	0.0	
2021-02-09	2022-01-21	50.31	5.0	C	0	228	0.0	
2021-02-10	2022-01-21	51.2	5.0	C	0	228	0.0	
2021-02-11	2022-01-21	51.1	5.0	C	0	228	0.0	
2021-02-12	2022-01-21	52.4	5.0	C	0	228	0.0	

Expiry dates in Jan 2022 but multiple days of massive trading volumes and successive decreases in open interest. Why are so many contracts being exercised when the have 1 year left till expiry?? Thousands of contracts were opened and immediately exercised. No interest in these calls or changes in open interest once suspicious activity stops.



date	option_expiration	stock_price_close	strike	call_put	volume	open_interest	oi_diff	naked_short
2021-02-05	2021-02-05	63.77	2.5	C	0	0	-2.0	
2021-01-20	2021-02-12	39.12	2.5	C	0	0	0.0	
2021-01-21	2021-02-12	43.03	2.5	C	0	0	0.0	
2021-01-22	2021-02-12	65.01	2.5	C	0	0	0.0	
2021-01-25	2021-02-12	76.79	2.5	C	0	0	0.0	
2021-01-26	2021-02-12	147.98	2.5	C	0	0	0.0	
2021-01-27	2021-02-12	347.51	2.5	C	0	0	0.0	
2021-01-28	2021-02-12	193.6	2.5	C	5	0	0.0	1
2021-01-29	2021-02-12	325.0	2.5	C	0	5	5.0	
2021-02-01	2021-02-12	225.0	2.5	C	0	0	-5.0	
2021-02-02	2021-02-12	90.0	2.5	C	0	0	0.0	
2021-02-03	2021-02-12	92.41	2.5	C	0	0	0.0	
2021-02-04	2021-02-12	53.5	2.5	C	0	0	0.0	
2021-02-05	2021-02-12	63.77	2.5	C	12	0	0.0	1
2021-02-08	2021-02-12	60.0	2.5	C	1	0	0.0	
2021-02-09	2021-02-12	50.31	2.5	C	0	0	0.0	
2021-02-10	2021-02-12	51.2	2.5	C	0	0	0.0	
2021-02-11	2021-02-12	51.1	2.5	C	0	0	0.0	
2021-02-12	2021-02-12	52.4	2.5	C	0	0	0.0	

Some smaller every day instances of opening and exercising contracts immediately. Zero open interest at strike price of 2.5$ then a small number of contracts are opened and the exercised on the same day or the following day.

Using this labelling scheme I almost certainly include some activity that is not from hiding FTDs. I will also miss some FTD hiding activity because sometimes it can just blend in with normal options trading.

I expect the timing and distribution of classified Deep ITM calls to be accurate but the exact values to have some uncertainty - perhaps +/-20%.

Examine biases: 1006 of the 10204 labeled rows were identified as suspicious. This number is massively skewed!!! If our classifier labeled all data as 0's it could get an accuracy score of approx. 90%. This would not be very useful. This is an imbalanced classification problem. I won't go into all the details here but one way to help the model deal with this is to reweighs the training set so that there are an equal number of 1 and 0 labels to train on. I used a python tool box designed to help with the problem and a technique called BallancedBagging.

Develop features: This really is the secret sauce. I don't want to divulge too much but I used the information contained in the example tables and different combinations (interactions, ratios etc.). The basic idea is that all the relevant information that was used to label the data manually is contained in the different features for the algorithm to use.

Split data: I reserved 30% of the labelled data for the test set and used the remaining 70% for model training. *I purposefully did not choose to use a validation set because I cannot be fucked with tuning all the model hyperparameters for an extra 1-2% accuracy when I already have something useful*.

Train models: I used Scikit-Learn to train my models. Here is a good classification tutorial with example code. I used a recursive feature elimination and cross-validated selection (RFECV) of the best number of features. Of my initial 14 developed features 12 had statistical support for the model. The two poor performing features were removed.

For the model training I used the BallancedBagging-Classifier to help with data imbalances and wrapped this around 8 different commonly used classification algorithms.

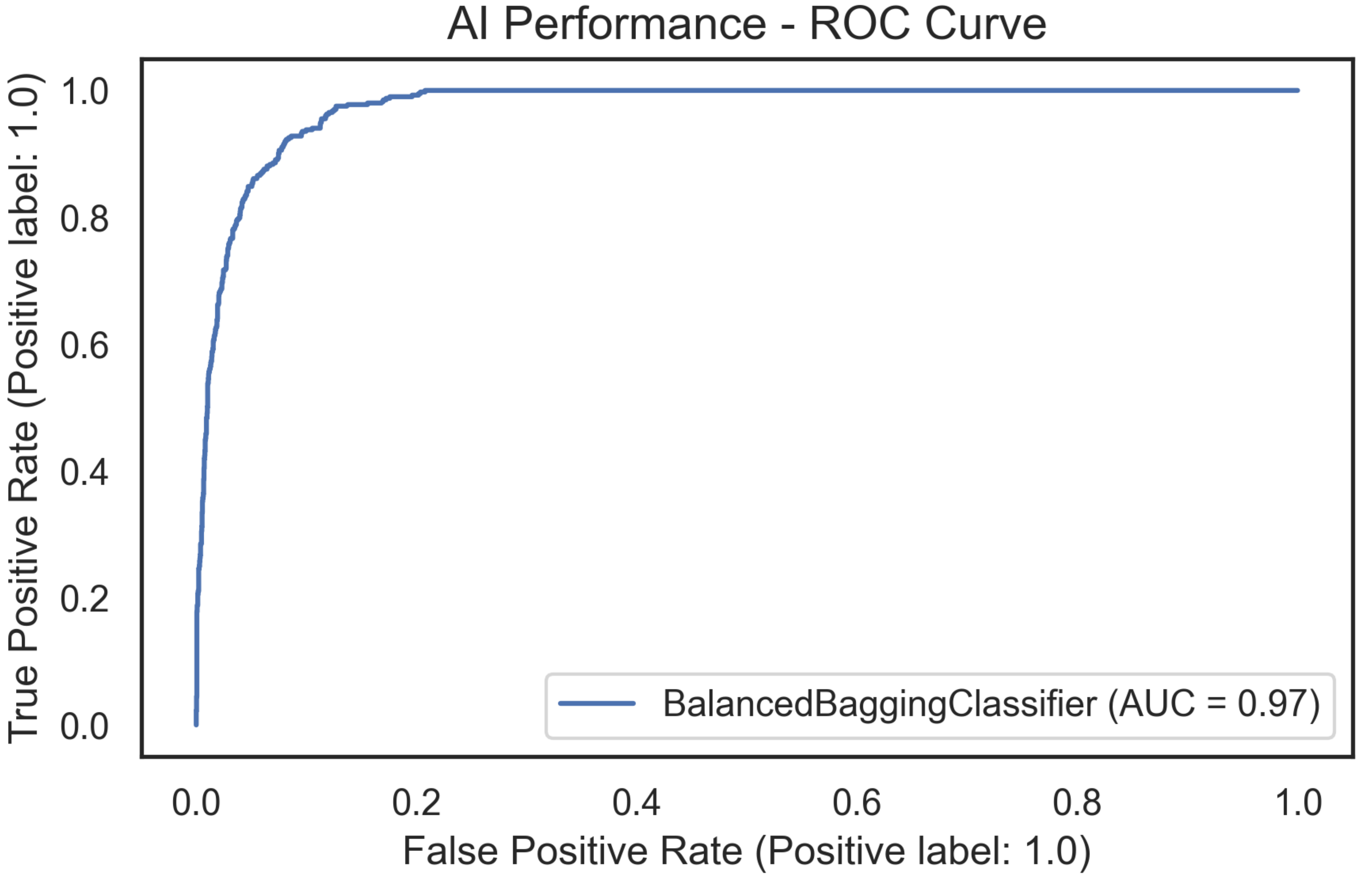
Test models: All testing was performed on the hold out test dataset. The algorithms had never seen this data before during training. Here are the accuracy scores for the different classifiers:

- Nearest Neighbors - Score: 0.88
- Linear SVM - Score: 0.79
- RBF SVM - Score: 0.88
- Decision Tree - Score: 0.91
- Random Forest - Score: 0.91
- Neural Net - Score: 0.89
- AdaBoost - Score: 0.91
- Naive Bayes - Score: 0.74

I used standard model tuning parameters as I wanted to avoid tuning all the different models. Because of this some models might perform poorly simply because they were not optimally tuned. Other models like the neural net might just need more data to perform optimally.

Model selection: All models performed reasonably well but I chose to use AdaBoost as it had the highest model performance of 91% (comparable to Decision Trees and Random Forests) and I like the theory behind the model.

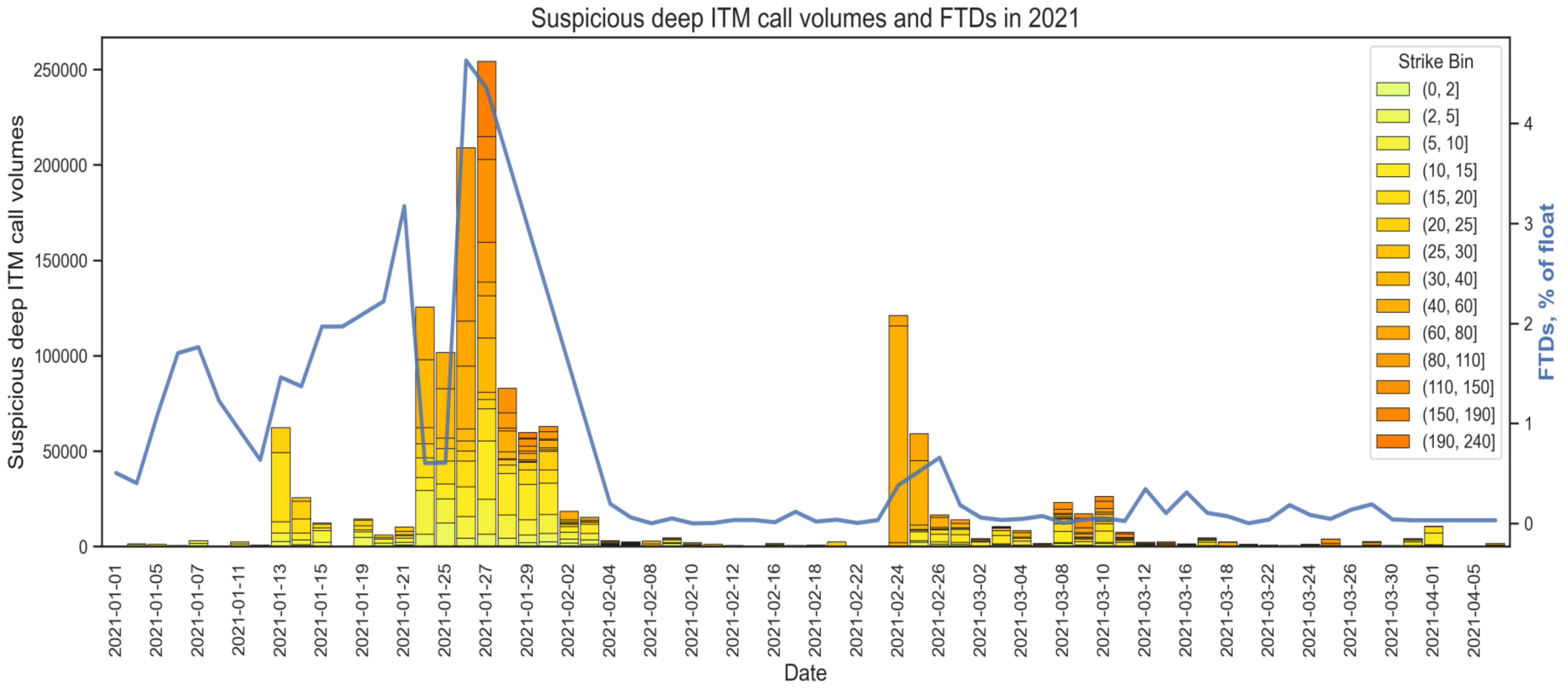
Here is a performance graph called an ROC-curve:



ROC-curve for the AdaBoost-BalancedBaggingClassifier. The model had very good performance on the test set with an AUC of 97%.

Model prediction: The trained AdaBoost-BalancedBaggingClassifier model was applied to all the other options data I had. I made sure to use the exact same strategy of generating features and normalisation before model predictions.

Explore the results: Ta-daaa!! We have some nice results automatically predicted for us by our trained classifier (AI):



AdaBoost-BalancedBaggingClassifier labelled Deep ITM call fuckery with overplayed FTD data. These plots were made using seaborn.pydata.org

Conclusions and potential biases

Lets start with the potential challenges and biases:

- Only suspicious data from 2021 was labelled
- Some normal options trading might be included in the manual labelling and automatic classification
- Some FTD hiding might not be picked up by the manual labelling and automatic classification because it is too well hidden in potentially normal looking volumes
- If any fuckery was happening at strikes >70% of share price I ignored them
- The model could be further improved with more tuning

Does any of this present a major challenge to the results? NO!

The classifier is still useful even if we cannot label the illegal activity with 100% accuracy. We might miss some, we might overestimate others. Overall the picture is still useful and we just have some uncertainty in the exact numbers we see. All models have uncertainty.

Could this work be improved upon? Of course. If someone could get more detailed data than end of day summaries we could remove a lot of the bias. We could relabel the dataset (but it would take much longer) and build an improved model. However 91% accuracy as compared to the best hand labelling scheme we have so far is pretty damn good.