The Clinical Trial Effect
A Machine Learning Framework for Predicting Returns with Clinical Trial Announcements

Roy Gabriel, Michele Orlandi, Riccardo Rossi, Michael Smith

December 14th, 2023

Abstract

In this paper we implement a machine learning (ML) framework, combining fine-tuned decoder-only large language models (LLMs) like PubMedBERT and FinBERT, with current well known tree-based machine learning models like XGBoost. This paper was inspired by the works of Budennyy et al. [3] yet our contribution extends beyond solely predicting reactions to clinical trial announcements. Our objective is to examine and predict the abnormal returns in pharmaceutical companies' stock prices due to the information contained in announcements regarding clinical trial developments. We fine-tune pre-trained LLMs on a corpus of almost 4000 unique clinical trial announcements on a text-classification task for positive, negative or neutral sentiment, achieving an F1-score of 0.86, and extract sentiment polarity scores for these announcements, as well as the respective companies 8-Ks using FinBERT. The results are aggregated and, together with technical indicators and financial information about the company, fed into XGBoost and Graph Convolutional Network models for a classification task of Normalized Cumulative Abnormal Returns. We achieve 53.2% accuracy and 0.48 F1-Score with XGBoost on a 3-category classification task.

1 Introduction

Pharmaceutical companies possess an interesting business model, often entrenched in lengthy, strict and especially expensive approval processes for their new drugs. A lot of money is poured into R&D to remain competitive in the market and explore new medical advancements. Hence, it is not surprising that their financial performance is heavily dependent on a combination of their current and developing drug portfolios. Because of the length and intricacy of FDA approval processes, clinical trial announcements - a series of FDA-mandated updates on the performance of a drug testing trial - become especially useful sources of information for investors, but also powerful catalysts for stock price movement. Smaller firms, with limited drug portfolios and lower market capitalization, will be particularly sensitive to the news reported in their own clinical trial announcements, as the approval of a new drug will hold a much higher weight in the success or failure of the company itself.

Within this context we frame our problem. Is it possible to accurately predict the intensity of the effect that the contents of a particular clinical trial announcement will have on the company's equity returns? The answer to this question involves combining Large Language Models and boosted trees in order to classify the *Normalized Cumulative Abnormal Returns* at some time t after the event. Throughout this paper we will illustrate our theoretical and practical implementation of our solution, as well as demonstrate a simple trading strategy based on our findings.

2 Data

We can divide our data into two broad categories, *Structured and Unstructured.* We describe them in detail below.

- **Unstructured Data**

    i. *Clinical Trial Announcements*: we collected a corpus of almost 4000 unique announcements for around 780 different pharmaceutical companies between 2009 and 2023, from `biopharmcatalyst.com`, which offers one of the most comprehensive datasets of clinical trial announcement. These observations consist of a few sentences (ranging from 1 to 5) summarizing the key results and outcomes of a specific trial. The summaries are organized by ticker, publication date, drug type and trial stage.

    ii. *SEC 8-K Filings*: For the same time span, we also collected quarterly 8-K Filings for each of the unique tickers in the clinical trial announcement dataset. We included this source as well, in hopes to gain further context and normalization for the sentiment scores. We scraped, cleaned and stored the data from the SEC's EDGAR platform, `sec.gov/Archives/filings`.

- **Structured Data**

    i. *Historical Market Data*: From `polygon.io` we retrieved twenty years of daily, and intraday, market data for each of the tickers in our universe.

    ii. *Historical Financial Data*: from The University of Pennsylvania's *Wharton Research Data Service (WRDS)* we downloaded historical financial ratios and metrics in order to provide insights into the financial health of the companies throughout our models.

3 Implementation

Before we dive into the technical specifications of our implementation, let us show a conceptual summary of the data flow and the logic in our overall model.

3.1 Sentiment Analysis

The first step in our project involves performing sentiment analysis on both clinical trial and 8-K text data. The goal is to extract *polarity scores* that indicate the level of *positivity, negativity or neutrality* in the provided text. The reason behind this solution is that the tone of these documents will have a direct impact on the magnitude of the "jump" or "dip" in a firm's stock price, thus they become crucial features in our downstream classification problem. The extraction of these features involves the *fine-tuning* and use of two LLMs, namely *PubMedBERT* and *FinBERT.* In the following sections we analyze our implementation of these models in greater detail.

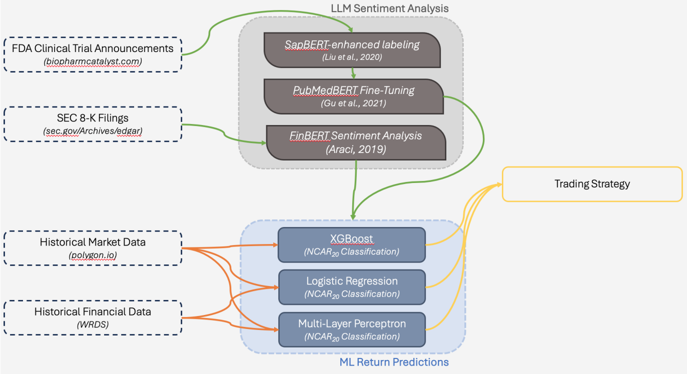


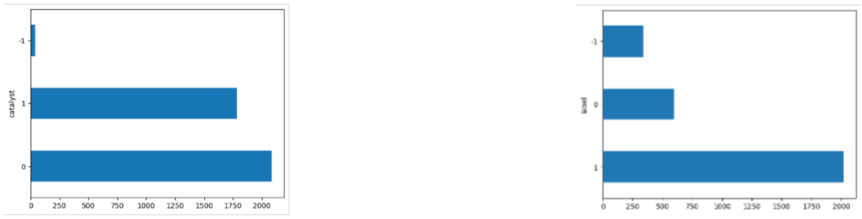
Figure 1: Model Pipeline

3.1.1 PubMedBERT Fine-Tuning

Fine-tuning refers to the process of training and testing a *Pre-Trained Language Model (PLM)* on a new set of text data, in order to ”increase” the knowledge domain of the model on a specific topic. In our case, we want our LLM to be able to understand the specific language contained in clinical trial announcements and accurately infer the tone of the announcement itself.

We decided to fine tune a PLM known as *PubMedBERT*, a BERT-based model that has been trained on a large corpus of medical data, this helps us by giving us an already ”knowledgeable” model in the medical space. To analyze SEC filings we used *FinBERT*, another PLM trained on financial text instead, in order to better capture the sentiment stemming from regulatory filings, where the language is certainly more standardized and appropriate to the financial domain rather than the medical one.

As any machine learning exercise, fine-tuning a model requires an objective function, in our case it is *text classification*, the ability of correctly categorize a piece of text, of varying length, into two or more categories. Akin to any other supervised learning problem, there is a need of ground truth labels for the text data. Given the size of our dataset and the nuance in the announcements text, we decided to follow a simple approach. Clinical trial announcements are often very poignant, they use specific words with unequivocal meaning, therefore we started with an initial guess of words that convey positive or negative sentiment (”approve”, ”meet” and ”halt”, ”fail”, ”stop” respectively), and followed a rule based approach, in which text containing any of these words would be labelled as positive or negative. The initial labeling, as shown in Fig.2a, shows a predominance of neutral and positive labels, allowing much room for improvement. We thus chose to employ another language



(a) Initial Labeling

(b) *SapBERT*-enhanced Labeling

model *SapBERT* to extract keywords of varying length [1;3], that would be able to optimize our ground truth labeling. The results of this methodology can be observed in Fig.2b. The number of neutral-sentiment text has significantly decreased, and we are able to detect a significantly larger proportion of unfavorable announcements. Despite the large imbalance between labels, the results are not surprising. As observed from the figure below, our dataset contains a vastly greater proportion of late-stage and approved trial observations, therefore it makes sense that the majority of the observations will have a positive label.

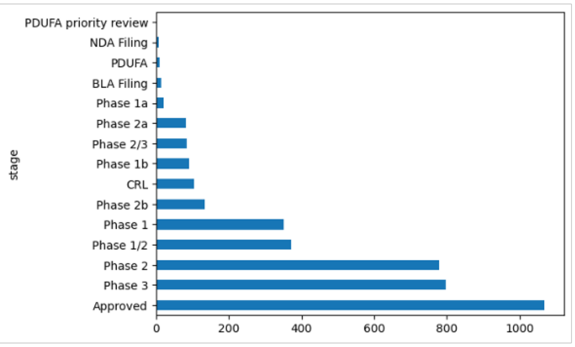


Figure 3: Clinical Trial Stages

The process of fine-tuning our PLM on our catalyst dataset was computationally expensive but rewarding. The hyperparameters used during training can be seen in table 1.

Hyperparameters	**Value**
Learning Rate	1e-06
Train Batch Size	64
Eval Batch Size	8
Seed	42
Optimizer	Adam
Epochs	1000

Table 1: Hyperparameters used during training of PLM on catalyst dataset.

After testing different iterations, models, and initializations, the *SapBERT* in conjunction with *PubMedBERT* [6] yielded the optimal results (F1 Score: 86.42 %). The model can be accessed **here**. The loss function used was the cross entropy loss, which is typical for classification tasks. Other model results can be seen in table 2. Other models include standalone *PubMedBERT* [4], *BioBERT* [5], and *SciBERT* [2].

3.1.2 Polarity Scores

After the fine-tuning process, the optimal PLM (*SapBERT* + *PubMedBERT*) was used to predict the sentiment of the catalyst dataset. Further, *FinBERT* [1] was used to predict the sentiment on 8-K filings for the given tickers associated with the clinical trial announcements. Since there

Model	F1 Score (%)
PubMedBERT + SapBERT (Liu et al., 2021)[a]	**86.42**
PubMedBERT (Gu et al., 2020)[b]	70.14
BioBERT (Lee et al., 2020)[c]	66.74
SciBERT (Beltagy et al., 2019)[d]	61.55

Table 2: Comparison of different PLM fine-tuning results on catalyst dataset.

[a] https://huggingface.co/cambridgeltl/SapBERT-from-PubMedBERT-fulltext.

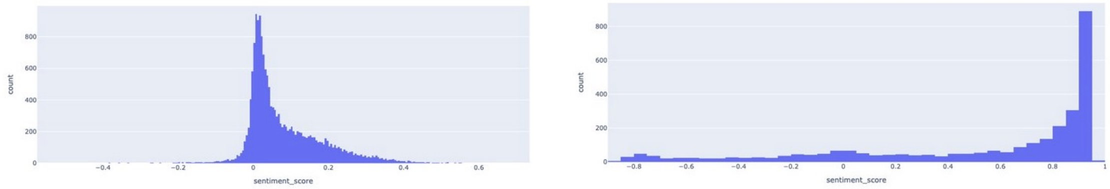
[b] https://huggingface.co/microsoft/BiomedNLP-PubMedBERT-base-uncased-abstract-fulltext

[c] https://huggingface.co/dmis-lab/biobert-v1.1.

[d] https://huggingface.co/allenai/scibert_scivocab_uncased.

were no labels with the 8-K filings, we assumed that the outputs from the *FinBERT* model were trustworthy.

During the process of predicting the sentiment scores, we were limited to only 512 tokens. Since some of our clinical trial announcements and especially our corpus of 8-K filings contained more than 512 tokens, we split the tokens up into chunks containing no more than 512 tokens each. This implies that our last chunk may contain less than 512 tokens. We applied transformations on the chunks, such as padding and reshaping the tensors, to be successfully passed to the BERT models. Since there were three class labels (0, 1, 2 - BEARISH, NEUTRAL, BULLISH), the output was in the shape of $(c \times 3)$, where c represents the chunk size. For example, if the original text was split into 3 chunks, each row would represent the different chunk and each column would represent the probability of belonging to each class label. This was obtained using the softmax activation function. After taking the average across the chunks, the output is in the form of (1×3), where each value represents the probability of belonging to each class label. In order to obtain a single polarity score, we multiplied that vector with the vector $[-1, 0, 1]^\intercal$ to obtain a compounded sentiment score in the range $[-1, 1]$. The closer the value to 1, the more positive or BULLISH the sentiment is. The closer the value to -1, the more negative or BEARISH the sentiment is. The closer the value is centered around 0, the more neutral the sentiment is. The distribution of our results for each of the datasets can be seen in Figures 4a and 4b.



(a) Compound sentiment distribution on 8-K filings. (b) Compound sentiment distribution on catalyst.

3.2 Features

In addition to the polarity sentiment scores, we added other features ranging from technical analysis to fundamental ratios of the company, thinking these would add valuable information to our predictive models.

3.2.1 Response Variables

For our response variables we chose to predict several metrics for the period starting with the catalyst date and ending with 20 days following the announcement. These metrics were *NCAR*, *Return*, *Volatility*, *Volatility - Volatility of Benchmark* and they were then terciled to create a classification model when predicting. We will often refer to the *NASDAQ Biotechnology Index (NBI)* to refer to our benchmark when discussing the response variables for our classification.

The *NCAR* and *Return* values were used to test the features with the ability of explaining stock returns following a clinical trial announcement. This was the main motivation of the modeling to create a trading strategy around the success of the classification. The *NCAR* is calculated using equation 3 and is used in contrast to the *Returns* in order to take into account the market or benchmark direction at the time of the clinical trial announcement as we want to predict the stock's reaction relative to current market conditions.

$$AR(t) = R(t) - BMR(t) \tag{1}$$

$$BMR(t) = \text{Benchmark Returns} \tag{2}$$

$$NCAR_T = \frac{\int_0^T AR(T)dt}{\int_0^T BMR(T)dt} \tag{3}$$

During our exploratory analysis, we found that there seemed to be an increase in volatility as the sentiment score for the clinical trial announcement tended towards the minimum and maximum groupings. The most neutral sentiment scores had the least amount of volatility. Because of this relationship, we decided to additionally test realized volatility metrics for the period following the catalyst event.

The *Volatility* metric is the volatility of the stock returns and the *Volatility - Volatility of Benchmark* metric is the volatility of the stock returns minus the volatility of the benchmark. Similar to the *NCAR* calculation, this metric was used to understand the stock movement relative to the benchmark.

3.2.2 Sentiment-based Features

As discussed in the previous section, we used catalyst texts and 8k texts as the two features representing the sentiment. These features are used to capture the sentiment of the stock surrounding the event with the hopes that this sentiment will be reflected in the stock returns and stock volatility in the period to follow. The correlation matrix with the two sentiment features and four response variables is given below in figure 5.

In reference to figure 5, there seems to be very little correlation between the response variables and the sentiment scores for both the *sec_score* and *bio_score*.

3.2.3 Technical Analysis

Regarding the technical analysis features, we chose a variety of these to capture the short trend movement of the stocks in order to possibly give our model useful information for the prediction of both NCARs and volatility. All the features that mention 'day', for instance 5 day average, implies that we calculated the statistic a certain number of days before the catalyst date.

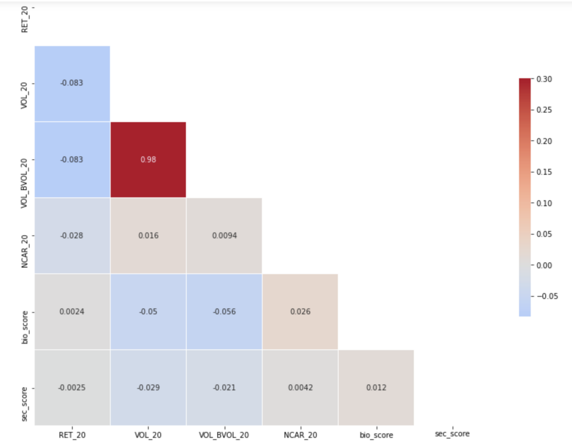


Figure 5: Sentiment Feature Correlation

The first set of features we chose are short term average and standard deviation minus long term average and standard deviation. Particularly, we chose 5 day average closing price minus 10 and 30 day average closing price, and 5 day standard deviation of closing price minus 10 and 30 day standard deviation of closing price. The reason we chose these particular features is because we believed that a high short term movement in the price could influence the price movement following the catalyst date. This is valid for both the differences of averages and standard deviations.

The second set of features are the 5 day average and standard deviation of returns. Again, these features are used because we believed that a short term movement in the stock price before the catalyst date could influence the NCAR and volatility following such date. Particularly, an up movement in the returns and volatility could signal confidence in the fact that the trial is positive for the company.

The third set of technical analysis features we selected are 5 day standard deviations of volume and transactions, together with the volume and transactions the day before the catalyst date. Even if these two features are poised to be highly correlated with the previous ones, we wanted yet another feature that could signify the drawing of attention towards a company in the days previous to the announcement.

The last technical feature we selected is the average true range, or ATR. ATR is a technical analysis indicator used to measure market volatility and does not provide directional information (up or down trend), but rather it gives a measure of the degree of price volatility. The "true range" of a security for a given period is the greatest of the following three values: (i) Current High minus the Current Low, which measures the intraday range of the prices, (ii) Current High minus the Previous Close, whose value signifies a gap up or down if this value is greater than the current high minus the current low and (iii) Previous Close minus the Current Low - similarly, if this value is the greatest of the three, it indicates a significant gap from the previous day's close. The ATR is then calculated as a moving average of these true range values over a specified period, and we chose 10 days before the trial.

3.2.4 Fundamental Analysis

We thought that technical analysis could be limited to the fact of just showing short term information of the company, so we added some longer term indicators to aid our predictions. Such longer term indicators are typically chosen from the set of financial ratios that can be observed up to a quarterly basis for each company in their financial disclosures. In our case, we chose to select the financial ratios from the closest yearly financial disclosure in time before the catalyst. The four financial ratios we chose to select are Return on Equity (ROE), Operating Margin, Net Margin and Debt Ratio.

ROE is a measure of a company's profitability relative to shareholders' equity. It is calculated by dividing net income by shareholders' equity. ROE shows how effectively management is using a company's assets to create profits. Generally, a higher ROE indicates more efficient use of equity.

Operating margin is a profitability ratio that measures what percentage of a company's revenue is left over after paying for variable costs of production, such as wages and raw materials, but before paying interest or tax. It's calculated by dividing operating income by net sales. This margin indicates how much profit a company makes on a dollar of sales before interest and taxes.

Net margin, also known as net profit margin, is the ratio of net profits to revenues for a company or business segment. It shows how much of each dollar in revenues is translated into profits after all expenses are paid. Calculated as net income divided by revenue, this ratio is a good indicator of the overall profitability of a company.

The debt ratio is a financial ratio that measures the extent of a company's leverage. It is calculated by dividing total liabilities by total assets. The debt ratio shows what proportion of a company's assets is financed through debt. A high debt ratio might indicate a high degree of leverage, which could suggest higher financial risk.

Each of these metrics offers insight into different aspects of a company's financial health and performance. ROE focuses on shareholder equity efficiency, operating margin and net margin provide a view into profitability, and the debt ratio offers perspective on financial leverage and risk. Together, these indicators can give a comprehensive view of a company's financial status and, in our minds, could provide some predictive ability to our model on the outcome of the catalyst.

As a last feature, we also added the size of the company in terms of market capitalization. Our idea was that the size of the company would be a very good predictor for the magnitude of the price change following the catalyst result, with the smaller companies experiencing the highest volatility.

3.3 Return Predictions

3.3.1 Logistic Regression

The core idea behind Logistic Regression lies in modeling the probability of a particular outcome occurring. Unlike linear regression, Logistic Regression utilizes the logistic function (also known as the sigmoid function) to squash the output into a range between 0 and 1, representing probabilities. Logistic Regression estimates the relationship between the independent variables (features) and the log-odds of the target variable. The model provides coefficients for each feature, offering insights into their impact on the likelihood of the outcome. Interpret-ability is a notable strength of this model. Because we are doing multi-class classification, we are using One-vs-Rest(OvR) to classify the groups accordingly. Moreover, Logistic Regression is less susceptible to overfitting, making it an excellent choice when dealing with limited datasets. It also provides probabilistic predictions, enabling users to set custom decision thresholds based on the specific needs of the problem. Since

Logistic Regression serves as a cornerstone in the classification domain, we chose to use it as a useful benchmark for our analysis in order to compare other, more complex models.

We have performed grid search for our logistic regression using the hyperparameters given in table 3.

Hyperparameter	**Options**
C	0.001, 0.01, 0.1, 1, 10, 100
penalty	'none', 'l2'

Table 3: Optional Hyperparameters for Logistic Regression

After running the grid search for the logistic regression model, table 4 represent the optimal hyperparameters for our logistic regression model in each of the four response variables. The given hyperparameters affect the model's regularization. The C parameter is the inverse of the regularization strength which increases regularization the smaller the value of C. The *penalty* hyperparameter adjusts which regularization method used.

Variable	***C***	***penalty***
NCAR	0.1	'l2'
RET	1	'l2'
VOL	100	'l2'
VOL - BVOL	10	'l2'

Table 4: Optimal Hyperparameters for LR

3.3.2 XGBoost

XGBoost stands as a formidable force in the realm of classification algorithms, often celebrated for its efficiency and predictive power. Built on the foundation of gradient boosting, XGBoost is particularly well-suited for both binary and multi-class classification tasks. At its core, XGBoost leverages an ensemble of decision trees, combining their outputs to form a robust and accurate predictive model. The "boosting" component involves sequentially building trees, with each subsequent tree correcting errors made by the previous ones. This iterative refinement results in a highly adaptive and powerful model. XGBoost's strength lies in its ability to handle complex relationships within the data, capturing intricate patterns that might be challenging for other algorithms. It introduces regularization terms in its objective function, mitigating overfitting and enhancing generalization performance. Additionally, XGBoost provides a range of hyperparameters that enable fine-tuning for optimal model performance.

We have performed grid search for our XGBoost Classifier using the hyperparameters given in table 5.

After running the grid search for the XGBoost model, table 6 represent the optimal hyperparameters for our model in each of the four response variables. The given hyperparameters affect the model's complexity and optimization rate. The *learning_rate* parameter specifies the rate at which the model converges to a solution. The *max_depth* parameter specifies the maximum depth of the trees in the boosting process. The *n_estimators* parameter specifies the number of boosting rounds.

Hyperparameter	**Options**
learning_rate	0.01, 0.1, 0.2
max_depth	3, 5, 7
n_estimators	50, 100, 200

Table 5: Optional Hyperparameters for XGBoost Classifier

Variable	***learning_rate***	***max_depth***	***n_estimators***
NCAR	0.01	3	100
RET	0.01	3	50
VOL	0.01	3	50
VOL - BVOL	0.1	3	50

Table 6: Optimal Hyperparameters for XGB

3.3.3 Graph Convolutional Neural Network

Pharmaceutical companies and their clinical trials are inextricably linked. The success or failure of a certain drug has operational and financial ramifications not only for its company, but also for those competitors with similar products in development or in the same industry, and several empirical studies demonstrate this connection through studies of returns and volatility around clinical trial announcement dates.

To incorporate these relationships and effects into our predictions, we decided to implement a *Graph Convolutional Neural Network (GNN)*, a type of network able to capture complex relationships explained by a graph. Firstly, we built an instance of an *Undirected Graph* $G = (V, E)$, where each vertex $v \in V$ corresponds to a unique clinical trial announcement, and each edge $e \in E$ represents the influence of a trial over another. To connect the vertices we used a couple straightforward rules: (1) if two clinical trial announcement are from the same company, and if the announcements have been published within a calendar year of each other, (2) if they are at the same stage in the clinical trial (i.e. Phase 1, Phase 2...), or (3) if two trials are focused on the development of a drug aiming at curing the same disease. The training set for this model included 1832 nodes or unique observations of clinical trial announcements (i.e. vertices), with around $734,000$ edges. The testing set for this model included 459 nodes or unique observations of clinical trial announcements (i.e., vertices), with around $37,000$ edges. In figure 6 we can observe a visual example of our graph for a sub-sample of training observations with 100 nodes and around $5,000$ edges.

4 Results

4.1 Classification Results

The results for each response variable and for each model were compiled and are given in table 7. The classification metrics that will be discussed are *AUC*, *Accuracy*, *Precision*, *Recall*, *F1 Score*.

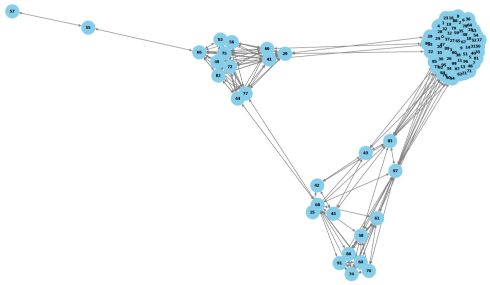


Figure 6: Undirected Graph of Clinical Trial Announcements

Variable	Model	AUC	Accuracy	Precision	Recall	F1 Score
NCAR	GNN	0.602	0.425	0.399	0.405	0.398
	LR	0.607	0.412	0.458	0.431	0.427
	XGB	**0.640**	**0.532**	**0.498**	**0.481**	**0.478**
RET	GNN	**0.542**	0.333	0.314	0.354	0.273
	LR	0.542	0.383	**0.384**	**0.382**	**0.377**
	XGB	0.517	**0.405**	0.362	0.355	0.345
VOL	GNN	0.554	0.379	0.314	0.368	0.305
	LR	0.574	0.427	**0.434**	**0.422**	**0.426**
	XGB	**0.575**	**0.429**	0.429	0.419	0.418
VOL - BVOL	GNN	0.560	0.401	0.409	0.407	0.305
	LR	0.628	0.473	0.459	0.481	0.459
	XGB	**0.629**	**0.508**	**0.481**	**0.506**	**0.484**

Table 7: Classification Results

4.1.1 Response Variables

When interpreting the results given in table 7, it is important to consider the utility of our modeling efforts. Because we want to implement a trading strategy where we would long the stock if we predicted the high group and short the stock if we predicted the low group, we decided that it would be best to separate the classification efforts into three categories so we can provide a buffer even if the response variable is miss-classified. Therefore, in the overall success of our strategy, the measure of *Accuracy* may not be the most practical measure to consider if our models are just really effective at classifying the neutral class. Because of our planned approach, we will use *F1 Score* to judge the efficacy of each model to be inline with our end objectives.

Based on the results given in table 7, we were most successful in classifying the *VOL - BVOL* response variable with the *XGBoost Classifier* based on the *F1 Score* metric. For the *VOL - BVOL* variable we achieved solid results across each metric with the *Recall* having the most significant difference between response variables with a value of **0.506**. This indicates, that relative to the other models, we had a much higher ratio of true positives to false negatives.

The *NCAR* response variable also performed well, and we will use this variable to implement our trading strategy. All of the metrics were quite similar to VOL - $BVOL$, but the $NCAR$ models had improved *Accuracy* with a value of **0.532** for the *XGBoost Classifier*.

When comparing the results of each response variable, it is interesting to note that the variables that were normalized against *NBI* in some way, such as $NCAR$ or VOL - $BVOL$, performed much better than the RET and VOL response variables. This gives some indication that it is significant to consider the current market conditions when making predictions based on the features we used. In large part, by comparing our response variables to a benchmark and then grouping these variables, we can isolate the stock responses from the current market conditions and achieve better results.

While we had hypothesized that we would have an easier time classifying volatility based variables, we found that there was very little statistical improvement between the *F1 Score* of the volatility based response variable and the return based response variable. It is clear however that we did have more success when predicting volatility of returns rather than the returns themselves.

4.1.2 Models

In regards to the models used, the best performer was the *XGBoost Classifier*. The results obtained from the *XGBoost Classifier* were consistently used as our best metrics when comparing the response variables. It is interesting to note however that *XGB* was no longer preferred in our models that predicted variables with low predictability such as RET. When there was less information we could use in predicting our response variable, the *Logistic Regression* model was preferred, most likely due to its simplicity.

Surprisingly, we achieved relatively poor results for the *GNN* model. This is most likely due to the complexity of the model and the overall lack of data points. Because our graph was quite sparse and we had few meaningful connections between nodes, we could not realize the benefit of adding features such as *Disease*, *Stage*, and *Relative CT Date*. We hypothesize that if we had more data points, we could achieve better results with our *GNN* than what is discussed in this implementation.

4.1.3 Trading Strategy

Once we ran the classification models cited above, we tested out a simple trading strategy to verify whether our work could lead us to have an edge with respect to the market. After classifying the returns (RET_{20}) and NCARs ($NCAR_{20}$) into 3 categories, the strategy we applied was to go long in the highest predicted category, go short in the lowest and not invest in the middle one. To calculate the results, we used the 20 day return following the catalyst date, therefore considering a holding period of 20 consecutive days.

Model	RET_{20} **Prediction (%)**	$NCAR_{20}$ **Prediction (%)**
GCN	1.06	-0.3
Logistic Regression	1.07	0.08
XGBoost	1.5	-0.06
MLP	0.3	-0.04

Table 8: Average returns of trading strategy using different models.

The table above (table 8) shows the average return in percentage over all the trades for the strategies with the different models and target variables. This would be the same return if we would have invested the same amount in every trade we made. As we can see, all the average returns are very small in absolute value, implying that our classification predictions were evenly spread between the classes. There is a slight improvement with the RET_{20} prediction relative to the $NCAR_{20}$, and this is strange since accuracy-wise the models performed better for the prediction of the latter. This could be due to the fact that $NCAR$ is not a good predictor for the returns, given that we calculated it with respect to a benchmark. Overall, the performance of our trading strategies is not convincing, but at least we manage to preserve the majority of the capital at our theoretical disposal.

5 Conclusion

Research and Development (R&D) in the United States, both in the public and private sectors, is a double-edged sword. The process itself, often lengthy and riddled with trial and error, is extremely undervalued by markets and investors, as it often is an expensive investment with delayed and oftentimes little returns. Pharmaceutical companies, smaller ones in particular, are especially affected by this, as most of their business is entrenched in R&D. Hence, in this project we set out to study how updates on clinical trials, in the form of FDA announcements, could impact the financial performance of their companies, and if it were possible to accurately measure and predict the magnitude of the impact, in the form of either returns or price volatility.

We built a comprehensive machine learning framework, beginning with the fine-tuning of BERT-based language models and sentiment analysis on a large corpus of text data, and ending with a series of graph or tree-based models leveraging market data and sentiment scores to predict NCARs or volatility. Our results were interesting and indicative of the complexity of the relationship between R&D and financial markets. Our models show that it is possible to improve predictive performance thanks to unstructured data, however their impact might be more representative given a different structure to the input data. We decided to forgo a time-series based approach because of the scarcity of clinical trial announcements compared to market observations, however, with access to more granular trial data it would be possible to better map the relationship between development success and returns. There is also room for improvement around *Graph Networks*, where a deeper understanding of the intrinsic relationship between pharmaceutical companies, perhaps through their industry expertise or geographical location, can help better capture the influence of certain clinical trials on their network.

In conclusion, beyond the technical applications of language models and their growing utility in quantitative finance, we believe this project explores an interesting challenge both for business and financial markets participants. In their brief and poignant nature, are clinical trial announcements enough for investors to assess the performance of their investment, for regulators to assess the success, safety and viability of a new drug, and for pharmaceutical companies to be transparent and incentivized to produce high quality products?

References

[1] Dogu Araci. Finbert: Financial sentiment analysis with pre-trained language models. *CoRR*, abs/1908.10063, 2019.

[2] Iz Beltagy, Kyle Lo, and Arman Cohan. SciBERT: A pretrained language model for scientific text. In Kentaro Inui, Jing Jiang, Vincent Ng, and Xiaojun Wan, editors, *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3615–3620, Hong Kong, China, November 2019. Association for Computational Linguistics.

[3] Semen Budennyy, Alexey Kazakov, Elizaveta Kovtun, and Leonid Zhukov. New drugs and stock market: a machine learning framework for predicting pharma market reaction to clinical trial announcements. *Sci. Rep.*, 13(1):12817, August 2023.

[4] Yu Gu, Robert Tinn, Hao Cheng, Michael Lucas, Naoto Usuyama, Xiaodong Liu, Tristan Naumann, Jianfeng Gao, and Hoifung Poon. Domain-specific language model pretraining for biomedical natural language processing. *CoRR*, abs/2007.15779, 2020.

[5] Jinhyuk Lee, Wonjin Yoon, Sungdong Kim, Donghyeon Kim, Sunkyu Kim, Chan Ho So, and Jaewoo Kang. BioBERT: a pre-trained biomedical language representation model for biomedical text mining. *Bioinformatics*, 36(4):1234–1240, 09 2019.

[6] Fangyu Liu, Ehsan Shareghi, Zaiqiao Meng, Marco Basaldella, and Nigel Collier. Self-alignment pre-training for biomedical entity representations. *CoRR*, abs/2010.11784, 2020.

6 Appendix

6.1 Links to Github repository and HuggingFace model and dataset

- GitHub
- HuggingFace Model
- HuggingFace Dataset

6.2 Additional Results

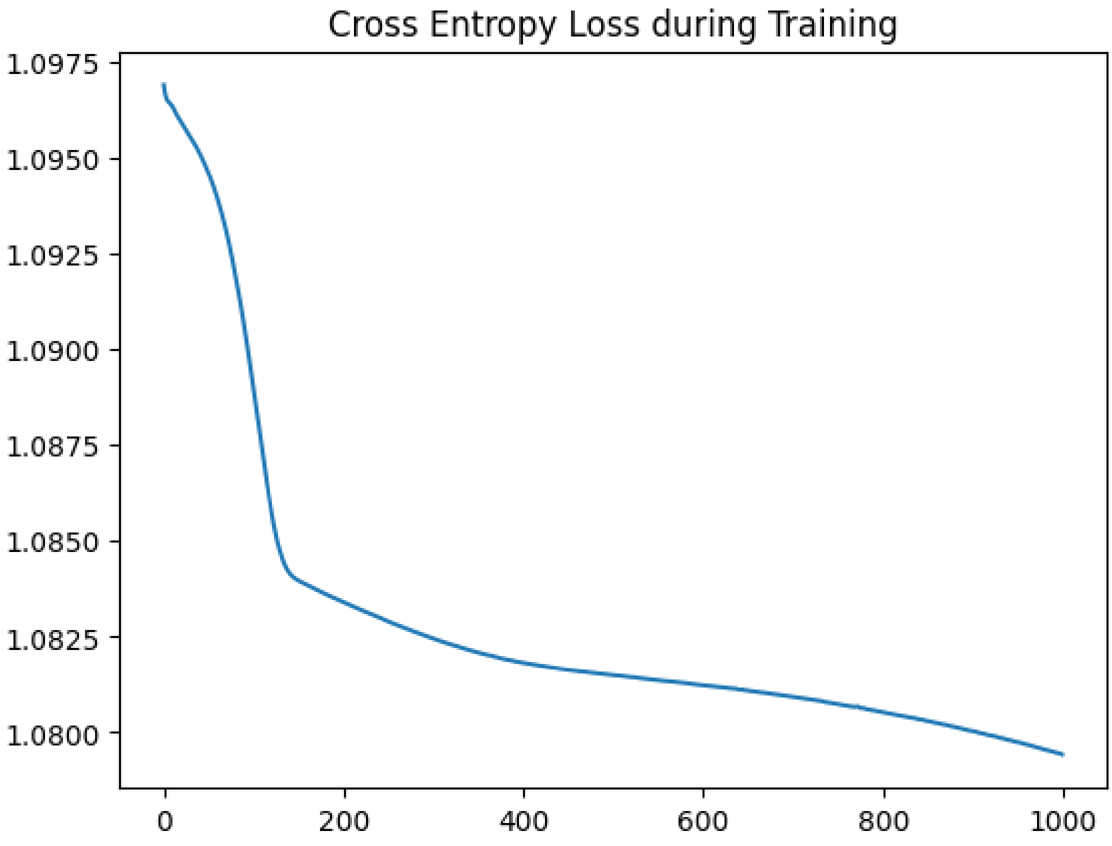


Figure 7: GCN Training Loss

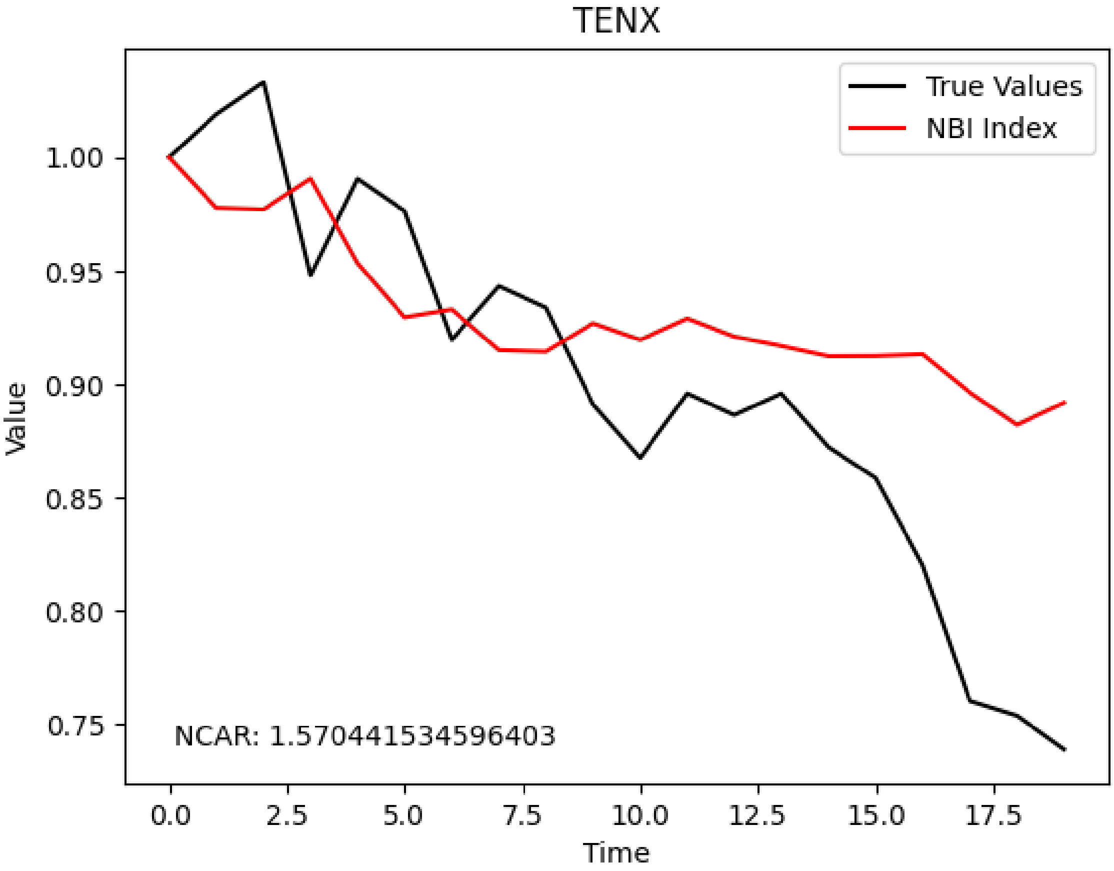


Figure 8: NCAR Plot

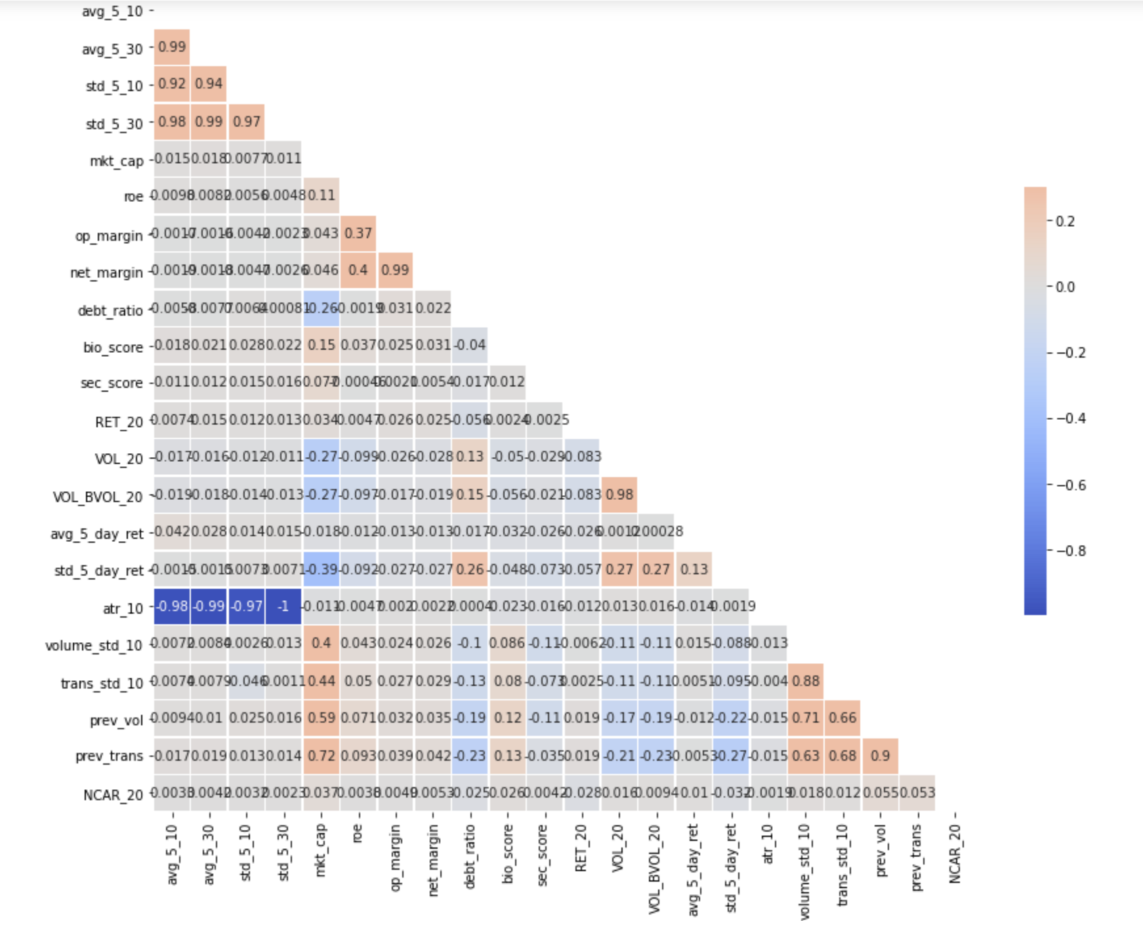


Figure 9: Total Features Correlation Matrix

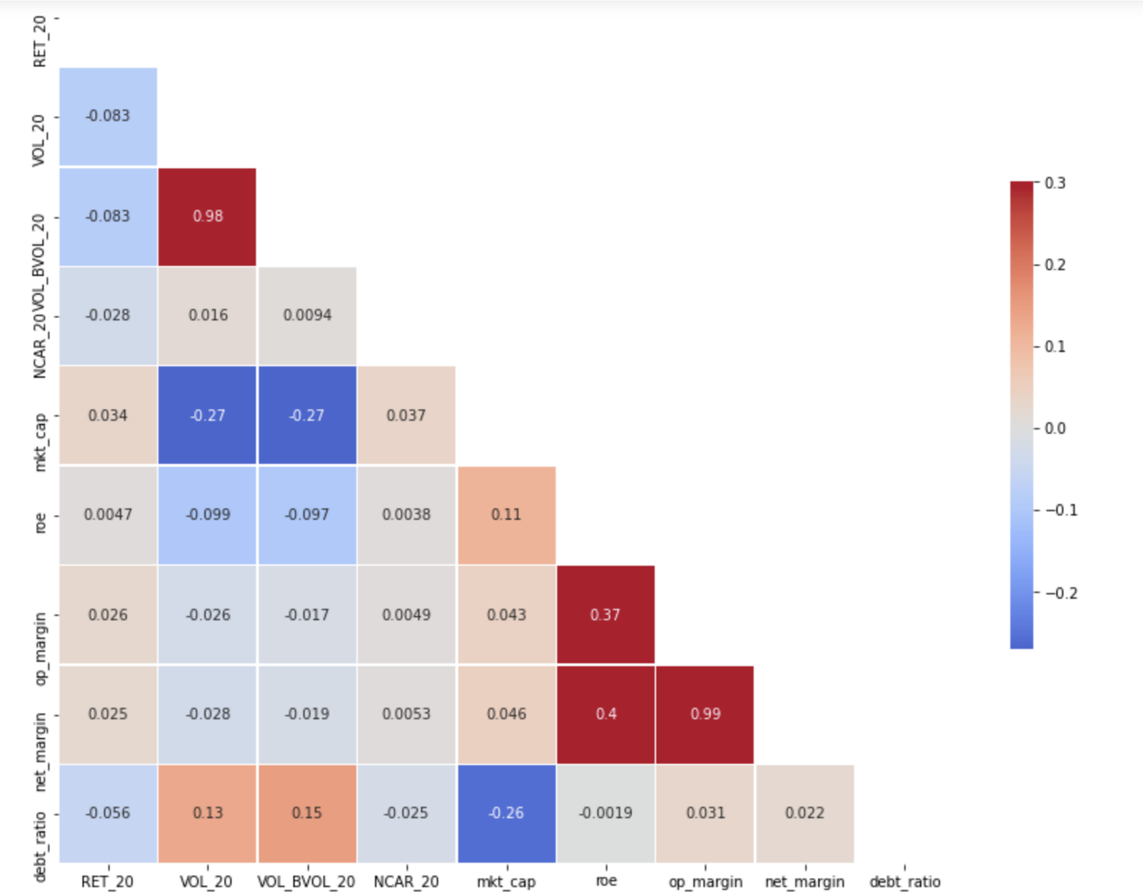


Figure 10: Fundamental Features Correlation Matrix

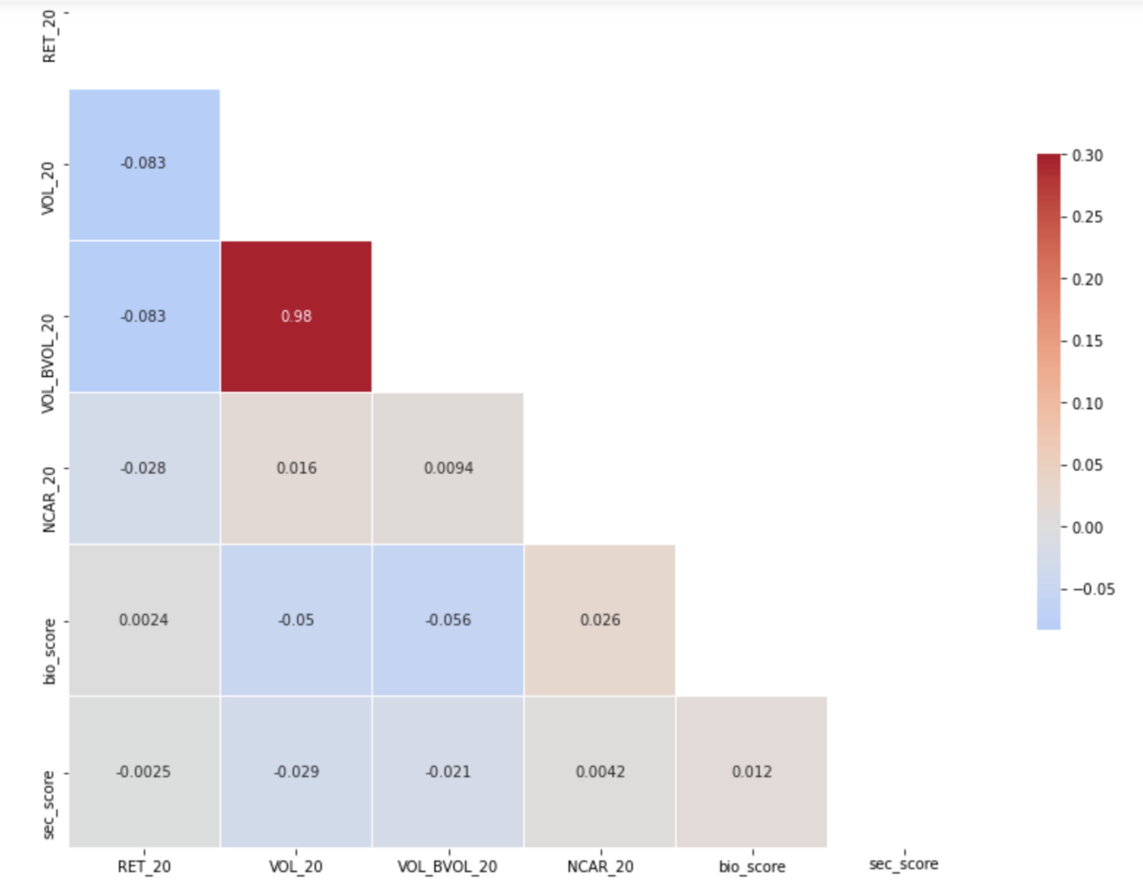
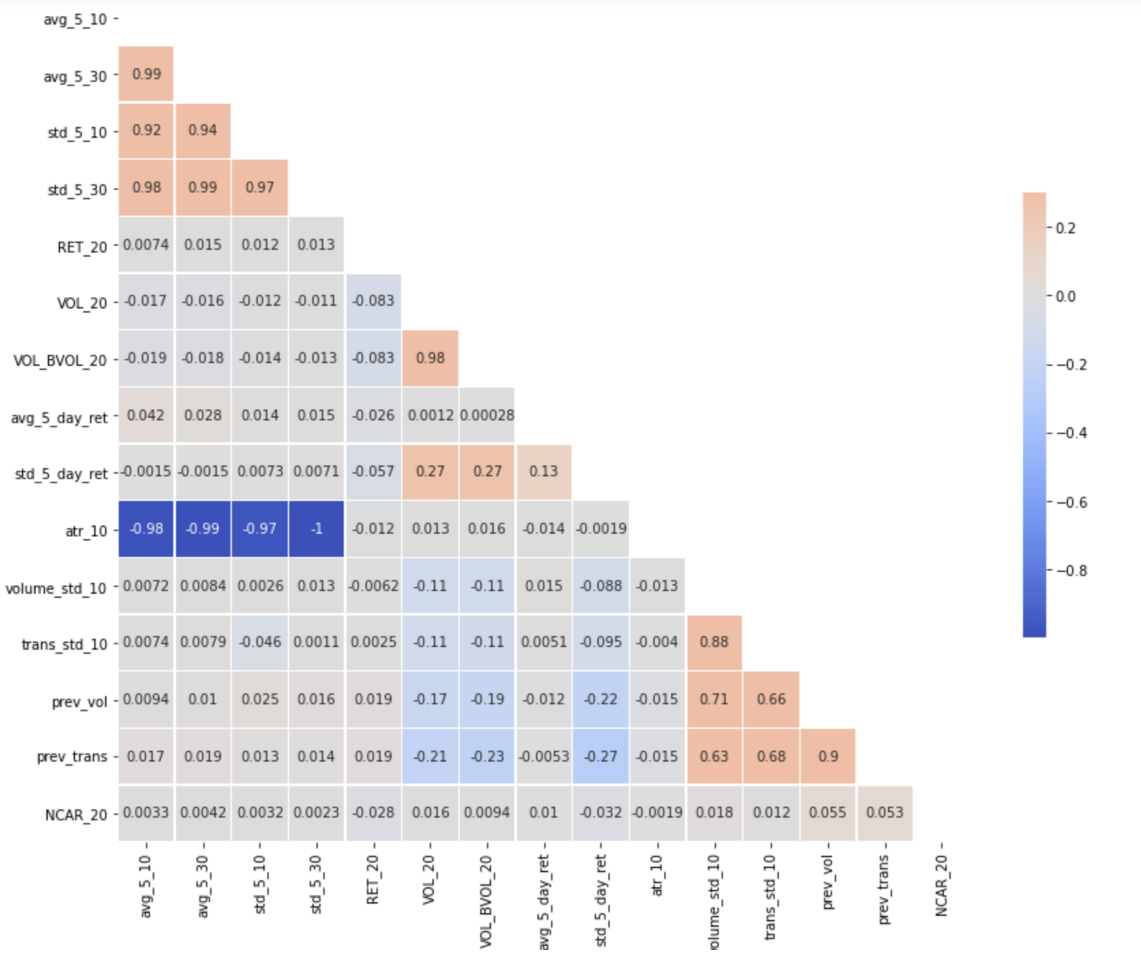


Figure 11: Sentiment Scores Correlation Matrix



	avg_5_10	avg_5_30	std_5_10	std_5_30	RET_20	VOL_20	VOL_BVOL_20	avg_5_day_ret	std_5_day_ret	atr_10	volume_std_10	trans_std_10	prev_vol	prev_trans	NCAR_20
avg_5_10															
avg_5_30	0.99														
std_5_10	0.92	0.94													
std_5_30	0.98	0.99	0.97												
RET_20	0.0074	0.015	0.012	0.013											
VOL_20	-0.017	-0.016	-0.012	-0.011	-0.083										
VOL_BVOL_20	-0.019	-0.018	-0.014	-0.013	-0.083	0.98									
avg_5_day_ret	0.042	0.028	0.014	0.015	-0.026	0.0012	0.00028								
std_5_day_ret	-0.0015	-0.0015	0.0073	0.0071	-0.057	0.27	0.27	0.13							
atr_10	-0.98	-0.99	-0.97	-1	-0.012	0.013	0.016	-0.014	-0.0019						
volume_std_10	0.0072	0.0084	0.0026	0.013	-0.0062	-0.11	-0.11	0.015	-0.088	-0.013					
trans_std_10	0.0074	0.0079	-0.046	0.0011	0.0025	-0.11	-0.11	0.0051	-0.095	-0.004	0.88				
prev_vol	0.0094	0.01	0.025	0.016	0.019	-0.17	-0.19	-0.012	-0.22	-0.015	0.71	0.66			
prev_trans	0.017	0.019	0.013	0.014	0.019	-0.21	-0.23	-0.0053	-0.27	-0.015	0.63	0.68	0.9		
NCAR_20	0.0033	0.0042	0.0032	0.0023	-0.028	0.016	0.0094	0.01	-0.032	-0.0019	0.018	0.012	0.055	0.053	

Figure 12: Technical Analysis Features Correlation Matrix