# Manipulation-Resistant Ethereum Price Prediction Using Large Language Models

Priya Roy

**Abstract**

The project focuses on building an Ethereum price prediction model along with eliminating all kinds of market manipulation effectively. This deals with the volatility of crypto market standards. There are traditional time series forecasting methods by using generalized ML models with some historical data that fall short in matters of unpredictability, where one cannot resist the data that are prone to external manipulations, hence resulting in wrongly predicted prices. To combat this situation, we implied here the use of Large Language Models (LLMs) power such as GPT-4o, along with a dataset combined to become a blockchain influence dataset where the on-chain data is gas fees. In the initial phase, the proposed approach was to use synthetic blockchain data, but expectedly, the synthetic data was unable to provide any predictions; hence, we switched to a fully historical version by collecting and merging the required components. We decided to keep the historical without on-chain data analysis for baseline comparison. We even tried the idea with the LSTM model across both datasets and evaluated based on the Mean Absolute Error (MAE), while for the actual project approach, we opted to fetch the predictions from the GPT-4o model through API calls using custom-formatted prompts to gather Next Day's Price Prediction for Ethereum. Ending the work with a promising evaluation from GPT-4o, along with the on-chain dataset pairing. The evaluation was not only satisfactory based on the MAE but also on the time taken to fetch the results without any errors encountered while running the predictions on LLM servers. With the other models and different datasets, pairings resulted in one or the other non-satisfactory results, consisting of unpredictable gaps. This approach not only improves accuracy but also makes transparent predictions, making its resultant behavior imply that the predictions are fully manipulation-resistant.

## I. Introduction

As we know, the crypto market is highly unpredictable, volatile, and susceptible to manipulation. We aren't even sure about where and what exactly the manipulations are coming from. The traditional time-series forecasting approach or the use of sentiment analysis falls short when performing in a volatile environment. These approaches are not considered workable enough with sudden market swings and misleading information. Our goal is to solve this existing issue where we are trying and testing different models and evaluating to decide on the best performance to outperform.

We are using primarily GPT-4o LLM to execute this idea with effective results. Unlike other ideas which highly depend on their predictions on external factors such as sentiments from social-medias this idea depends on Ethereum's historical blockchain dataset (where gas fee is an on-chain data, hence, a blockchain dataset). There are other on-chain data we could include to refine our predictions, but for now, we would only be focusing on the gas fee. Reasoning out why we use gas fees helps us to intentionally avoid social media's sentiment-driven manipulations towards data, thus resulting in transparency in our forecasting.

The project workflow involves GPT-4o to process daily Ethereum prices for 30 days to return the next day's prices as a prediction for that particular day. This is all carried over by using customized prompts and passing the prompts as a request using the GPT-4o self-generated API key. Now, the analysis revolves around how closely this approach predicts by evaluating the actual price of the very next day. The standard metric that calculates the error also plays a key role when we are evaluating Mean Absolute Error (MAE). These are the two factors of analysis used in our predictions currently.

Further talking about one more metric that was proposed earlier to step in further for better analysis and performance was analyzing derivative market behavior (derivatives here is – Ethereum in financial language). This could be done by applying a simulated call-option function. Here, the call option means an Ethereum holder will buy Ethereum at a fixed price in the future. Simulation means that from the call option, the holder can calculate or simulate their profit and loss for the future on their purchase or sale. Now, while moving forward in the project, we dropped these evaluation metrics since they are more effective while we are predicting a future price, whereas currently, we are evaluating our prediction accuracy on historical data, as to how closely we were able to predict, along with feeding into the goal of eliminating any kind of manipulations.

## II. Background Information

Ethereum is a widely popular cryptocurrency in the trading market, one of the capital sources of investment and returns for day traders. Now, where do the pros that we see as high-profit margins come in no time, there come the cons as well. Unlike other stock trading games, which are less likely to be affected by these external manipulations like pump and dump tactics (large holders bulk buy and sell), impulse buying (new traders), and other factors like bots inducing fake prices. All these factors involve high risks and make the trading game go completely wrong here. And traditional forecasting methods here are not resistant to these mentioned manipulations.

To get to work on this effectively, we implement in our project blockchain methods where we focus on blockchain data, which could preferably be gas fees, transaction volume, network activity, and transaction count as well. Currently, in our project, we are using the gas fee only for now. Diving a bit deep into the explanation and use of gas fee, we can compare it as if sentiments are manipulations, then gas fee acts as its resistance. We know that sentiment manipulations and the gas fee are taken as resistance because it is a direct indicator of on-chain activity or the demands of Ethereum. By considering gas fees as input data, we induce typical blockchain behavior, which is difficult to manipulate by any means. With the due course of the semester, the project kept on evolving as part of personal research using different approaches, comparisons, and datasets based on my understanding while moving further to better it with the help of faculty and cutting-edge questions to work on to refine the analysis. We started with custom-created LLMs, which would replicate GPT-4o, to transitioning into the approach of prompt-based API request calls.

When we talked about transitioning to GPT-4o LLMs there was an additional reason which was to work as expected here we would require large LLM servers which is not possible for a small-scale project with a personal system, hence this API approach was opted for, another reason was LLM would give textual responses to some of the predictions if it fails to do so sometimes with reasons, we just need to hold a place to bypass those textual responses so that it won't mingle up with the columns which are expected to have numerical responses. Also, GPT-4o API calls are paid, why only GPT-4o, and why not other LLMs like Deepseek or any other equivalent? GPTs have high-performance prediction engines running through their servers, which work throughout providing reliability, textual response consistency, increasing interpretability, and smooth working with volatile datasets. While other open-source LLMs may be cost-effective, they need further fine-tuning using additional layered models. And this can only be backed by applying some future analysis on working with other LLMs.

## III. Motivation

The original motivation behind the project idea was based on a research paper, which already included this approach of using GPT-4o LLM primarily, which was why this was the first and foremost to experiment on. The research paper that derived this influence is "Manipulation-Resistant Prediction Market Derivatives with Language Models". According to this research, an experiment was conducted where the Ethereum Polymarket (platform where Ethereum prediction contracts take place) was intentionally manipulated by investing a large sum of around $7million.Nowthishighvolumepumpisunacceptabletocryptotradingstandardsasitgetshighlyinfluencedtowardscreatingasen$

The proposed idea states to use generalized LLMs to generate fair predictions instead of depending on guesswork based on wrong data, which won't be easily fooled by this kind of manipulation. GPT-4 was expected to pull information from various sources like trading indicators, economic trends, and news, even social sentiments.

The goal is just to gather correct information, but with traditional forecasting or even just LLM without blockchain or structured Ethereum data like gas fee pulls on manipulated details as well, hence, here they used LLM with blockchain to filter out all the manipulations. Involving all these features makes sure to incorporate a trustworthy and robust prediction. These approaches can be considered to be practiced in the future, saving the crypto trade from collapsing as a result of manipulations.

Hence, this is how this project outline got inspired to carry out this approach to have a practical implementation experience as well as the blockers and how to solve them, like we did not use GPT-4 here since for the system in which is project was carried over was mostly compatible with GPT-4o (Omni) version as an alternative for a low manipulated predictions.

So, the project builds the foundation of an unexpected idea of implementing a GPT-based manipulation resistance prediction system while confirming with different experiments that it works, stepping further towards fairer crypto market trends.

## IV. Problem Statement

The price prediction for Ethereum is affected by market manipulation, wrong information, and its volatile nature. The techniques used to combat this issue are either sentiment analysis, LSTMs etc. This project will investigate the utilization of LLMs to mitigate the manipulation risk and achieve robust price predictions for Ethereum. It will also work on the challenge of using LLMs my calling Open AI platform API with customized prompts for better performance, which would be tough for external manipulations and yield a realistic performance.

## V. Proposed Solution

According to the initial proposed solution for this project's scope and purpose, we worked with a synthetic blockchain dataset along with a self-made LLM model which would somewhat behave like GPT-4o and had its results as a baseline comparison with another historical dataset without blockchain features and also experimenting with the LSTM model performance with both the datasets. Moving further, we even evaluated the simulations using the call-off function with a formula:

$$\text{Payoff}_i = \max(\text{ETH\_Price}_i - \text{Strike\_Price}, 0)$$

Fig. 1. Simulation of Call Option Payoff Function based on ETH Price

As we see in the graphs, there were no better predictions coming up stating it as a justified result. Hence, this particular approach was dropped since it lacked forecasting ability and was no longer continued in the project approach.

After diving deep into the forthcoming questionnaire based on the mid-term presentation, leading to more discoveries and techniques to be uncovered in which there could be a considerable solution relevant and based on concepts circulated in the mentioned research paper, related works, and online materials, we changed the technique of using the GPT-4o LLM model to the most of its ability for predicting the next day's prices based on the historical dataset with blockchain features emphasizing the importance of using LLM along with blockchain feature for dealing with

Now our final approach includes two major steps:

a) Data Engineering Layer: Here we collected the historical Ethereum (ETH) standard prices and historical on-chain metrics gas fee for a month from March 28, 2025, till April 29, 2025, with each day as one single data point. We cleaned and merged the dataset by converting the gas fee unit (Wei) to ETH for better readability using the formula, and calculated the gas fee per transaction. Post this, we scaled the data into a range of [0,1] to bridge the gap between high magnitude differences so that GPT would treat the price column in USD and the gas fee paid column in ETH as the same numbers, thus eliminating any issue caused by the model inputs.

b) GPT-4o Integration: Here, we first checked whether our system supports the GPT model, and GPT-4o was one of the latest ones supported, so we opted for that. Since it's a paid version, we had to make a pre-paid deposit of $10, thoughthepriceofeachAPIcallwasjust0.01$, and each call consumed 50 tokens for my set of datasets. We further created customized prompts and as accurately it could sound as possible. Then we made the code structure to make the API calls, where the responses came as the next day's predicted prices in the current day's row. We ran for both the standard historical dataset and the combined blockchain dataset based on Mean Absolute Error (MAE), where the blockchain dataset seemed to deliver better results with an MAE OF 52.90, while with the other dataset, the MAE was 54.70. We could see more stable and interpretable predictions for dealing with noisy data.

Now another is saying where it's believed that market manipulations nullify any predictions: It's true it surely does but just to the extent and to bridge the gap we are applying the solution here that is also why the reason the difference in both datasets is so low, but yet effective since we should always opt for a solution which lets us be close to accuracy even if it's slight enough.

Hardware Requirements: For calling the GPT-4o API

- Processor: Apple M3 chip
- RAM: 18GB
- Storage: 1TB

Software Requirements:

- Python:3.8.0
- Jupyter Notebook

- OpenAI Python SDK (¿=1.0.0)
- OpenAI API Key (Paid Version)

Libraries:

- Pandas,matplotlib, scikit-learn

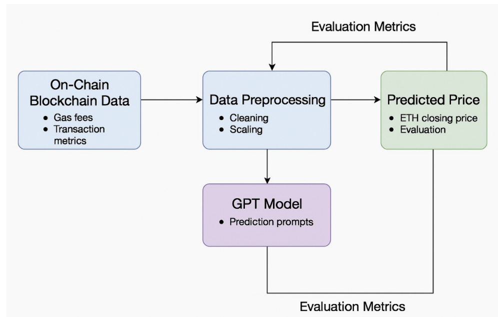


Fig. 2. Project Workflow

The proposed theoretical idea was successfully implemented, resulting in a practical application analysis of how much future predictions match with the historical dataset. We are not working here on actual future analysis. For example, if we need a prediction of tomorrow it is only possible if today’s pricing is available online while currently, this method only for now shows its ability to generate accurate next-day results based on already existing data which means it works on only retrospective evaluations just like a simple time-series forecasting but with better results.

## VI. Techniques Used

*A. Read and pre-processed the datasets by combining and aligning all the datasets.*

*B. Applying changes to the units of gas fees to a readable and comparable version, like converting it from Wei to ETH using a formula. .*

*C. Scaling – changed the targeted fields “Price”, “Next Day’s Predicted Price”, and “GasFees*$_{P}aid$”$intoafloattypeandinaconsisten$

*D. Adding customized prompts for GPT-4o. .*

*E. Calling GPT-4o API for having responses on “Next Days Predicted Price”.*

*F. Evaluating model performance by “Mean Absolute Error”.*

*G. Plotting for visualizations.*

## VII. Evaluation

In the evaluation phase, we will be talking about just the final results that we obtained throughout the project and the loopholes we discovered during the workflows. In this part, we will dive a bit deeper into the technical parts of the implementations.

Now to achieve this final goal we have proposed a two-step evaluation pipeline which would serve also as a baseline comparison. Re-iterating a few known facts and the reason behind using them were like unit conversions: to increase readability, date format alignment: to avoid GPT getting confused resulting in no predictions, scaling and normalizing the dataset: for effective predictions. The two-step evaluation measures are firstly, accurate prompts and then sending the API request. In response, we got the next day’s predicted price compared against the actual closing price for the next day. Secondly, the prediction comparison took place with the help of Mean Absolute Metrics (MAE) as one of the main evaluation metrics. The same approach is repeated across both datasets.

Now, talking about all kinds of challenges faced in evaluations. The evaluation with the blockchain dataset went really in just one single go without any blockcage or error. But without the blockage chain dataset which is also the standard historical data, multiple unexpected behaviors occurred causing great confusion in evaluating cause were not sure whether the obtained results were manipulation-free or not. What was happening was that GPT-4o was struggling to make predictions for Ethereum’s next-day prices. Initially, with each call, it would give messages saying it cannot predict. After a few seconds, we were able to see the predictions happening, but we also saw multiple “NaN” values, which is nothing but a string from GPT-4o, indicating it’s not able to predict for that row.

The interesting part is that with each call, there are different rows we can see with "NaN" values. Please note that the predicted values are not shuffling, but the "NaN" values are. Making it confusing to interpret which row was exactly affected and how to clear them. Please note, even though the **Next Day Predicted Price** column was set to send and receive only numeric values GPT still can send string values in them stating its inability to predict for those entries. Still, this was an expected behavior it's the shuffling that is causing uninterpretable behavior hence resulting in poor consistency. We tried to even use more accurate prompts, cleaning and re-cleaning the data, still didn't work at the end just to show case the plottings to still able to make a difference in performance for available data we dropped those rows with "NaN" values, and went ahead to calculate MAE and continue with the plottings.

### A. *Datasets and Variables*

- Standard Ethereum historical dataset
- Gas Price On-Chain Dataset
- Combined Blockchain and Price Dataset

The target variable was the Ethereum closing price for the next day. Other input variables involved and involved in our analysis were:

- Price: Standard historical Ethereum price in USD
- Gas price: (in Wei)
- Gas fees (in ETH derived as GasPrice ETH × 21,000)

### B. *Experimental Setup*

```
Given the Ethereum network data:
- Previous Closing Price (USD): 1799.24
- Gas Fees Paid (ETH): 4.2391696473e-05
```

Fig. 3. Sample Prompts

GPT's response were fetched in a new column Next Day Predicted Price, along with rows having string values were programmed to get dropped automatically.

### C. *Metrics Used, Results and Interpretation*

$$\text{MAE} = (1 / n) \times \Sigma\, |\text{actual}_n - \text{predicted}_n|$$

Fig. 4. MAE Formula

- n is the number of predictions
- actual is the real observed value at position n
- predicted is the model's predicted value at position n
- mod represents the absolute value

We decided to go with MAE since it gives more clarity in metrics which are there to be interpreted against the outliers, just completely suitable for numerical predictions.

Talking about the results, MAE for standard historical prices was 54.70 USD while that of historical blockchain datasets was 52.90 USD. This clearly shows the difference the presence of blockchain features(gas fee) made. Not only in evaluation metric but also in speed of getting the reliable output in time with no errors.

### D. *Data Visualization*

From the plotted graphs we made few of the basic observations like:

- When the market is less volatile in those time we could see stable alighnment in market trends.
- Sharp price bump and dumps invited unwanted errors.
- On comparing the working of both the datasets, on-chain data top as to perform without any blockage.

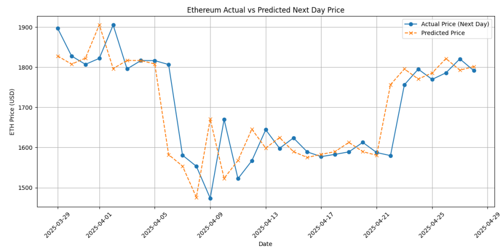


Fig. 5. Ethereum Historical BlockChain Dataset Plot

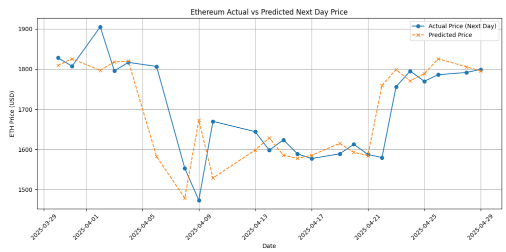


Fig. 6. Ethereum Standard Historical Dataset Plot

### E. Error Analysis

Now, talking about all kinds of challenges faced in evaluations. The evaluation with the blockchain dataset went really well in just one single go without any blockcage or error. But without the blockage chain dataset which is also the standard historical data, multiple unexpected behaviors occurred causing great confusion in evaluating cause were not sure whether the obtained results were manipulation-free or not. What was happening was that GPT-4o was struggling to make predictions for Ethereum's next-day prices. Initially, with each call, it would give messages saying it cannot predict. After a few seconds, we were able to see the predictions happening, but we also saw multiple "NaN" values, which is nothing but a string from GPT-4o, indicating it's not able to predict for that row. The interesting part is that with each call, there are different rows we can see with "NaN" values. Please note that the predicted values are not shuffling, but the "NaN" values are. Making it confusing to interpret which row was exactly affected and how to clear them. Please note, even though the Next Day Predicted Price column was set to send and receive only numeric values GPT still can send string values in them stating its inability to predict for those entries. Still, this was an expected behavior it's the shuffling that is causing uninterpretable behavior hence resulting in poor consistency. We tried to even use more accurate prompts, cleaning and re-cleaning the data, still didn't work at the end just to show case the plottings to still able to make a difference in performance for available data we dropped those rows with "NaN" values, and went ahead to calculate MAE and continue with the plottings.

### F. Lessons Learned

- Near accuracy in prompts means changing the prompts till we reach our desired results often works well.
- On-chain data adds resistance to market external manipulations, as here the gas fee increases the accuracy.
- LLM unpredictability returns a textual response as a verbose, uncertain reply instead of numerical values, irrespective of the column datatype
- Repeatability increases the learning of the model, resulting in proper predictions, meaning the more you repeat the API calls, the better results are possible, mostly, but depending on the correct dataset you are working only then is it possible

## VIII. Related Work

### A. Traditional Time-Series Forecasting

There are numerous machine learning models like Random Forests, LSTM, and ARIMA models. These are involved in cryptocurrency price predictions. Even after the success of these models, however, they rely on historical price data and are heavily dependent on sudden market manipulations.

### B. Blockchain Analytics and Market Derivatives

According to previous research, the potential for blockchain analytics is clearly demonstrated in predicting cryptocurrency behavior. However, there is very minimal use of Large Language Models for consolidation of information.

### C. Manipulation-Resistant Prediction Markets

Maximum work on manipulation-resistant markets has been based on top traditional market mechanisms and DeFi platforms. All these involve limited work of Large Language Models, which can be applied over these platforms mainly for predicting prices without any text-based analysis.

The idea of predicting cryptocurrency prices is not new—but the approach researchers and developers have taken to it has shifted dramatically in the past few years. One of the earliest papers that broke the mold was the research paper ”Blockchain Analytics and Market Derivatives” (arXiv:1810.06696). It addressed how blockchain-derived measures like gas price and transaction volume could correlate with financial derivative models. Essentially, it explored how blockchain data can be used to reflect trends in the wider markets, especially in trying to model call and put options. It assisted in creating a foundation of thought around crypto markets not as price charts but as data-rich systems. Another interesting contribution comes from Chaos Labs and their work entitled ”Edge Proofs: AI-powered prediction market oracles.” They posted an idea in their blog that involves using large language models (LLMs) to serve as oracles or trusted sources of information for prediction markets. They wished to insulate markets against manipulation by ensuring there are systems where players can’t easily trick the model into making bad predictions. This motivated our own efforts to build a manipulation-resistant forecasting system. Like them, we noticed Ethereum markets are exposed to huge trades, lies, and exploding gas prices. There is also an older school method, like in the research work entitled ”Cryptocurrency Price Prediction Based on ARIMA, Random Forest, and LSTM Algorithm.” It uses traditional forecasting techniques—ARIMA to capture statistical trends, Random Forest for decision tree models, and LSTM for learning sequences. Though excellent in certain stable environments, those models won’t account for manipulation or on-chain behaviors that actually distort projections in the cryptocurrency world. What’s new here is combining all of these concepts. We didn’t commit to vanilla machine learning or theory oracles—we used GPT-4 to generate real, next-day Ethereum price predictions, and we gave it real-world inputs like gas prices and transaction volume. We transcended theory into application: writing prompts, translating GPT responses, and testing results with metrics like MAE. While others have written about using LLMs or blockchain data, we actually deployed and experimented with a system that integrates both. In brief, this project doesn’t just study how AI can be used to support prediction markets—it illustrates a working implementation that combines blockchain analysis and language models to enhance better, stronger crypto prediction.

## IX. Timeline with Weekly Goals

### A. Week 1

**Study and review the research paper:** The week 1 we can spend studying various research papers relevant to existing prediction models and manipulation-resistant markets, which require work proposed over here. Furthermore, investigates bettering the LLM implementation and large-scale financial applications.

### B. Week 2

**Collect Ethereum standard historical price data and blockchain data studied it and, and preprocessed it:**In week 2, we collected Ethereum standard historical price data and blockchain data here and pre-processed it, along with sanitizing or cleansing and scaling the data for the LLM-based model.

### C. Week 3

**Experimented with API calls of GPT-4o:**In week 3, we created API keys and experimented a bit for it’s working.

### D. Week 4

**Train and evaluate the model based on accuracy:**In week 4, we trained and implemented OpenAI API call function with Ethereum standard price data and evaluated them against the predictions based block chain data.

### E. Week 5

**Simulate derivatives and test system performance**In week 5, we created manipulation-resistant derivatives, which are based on the Ethereums blockchain historical dataset prediction is about the next day's price predictions.

### F. Week 6

**Prepared the final report and presentation:**In the final week 6, we would be prepared the final project report documenting the updated steps, techniques and tools involved, methods posted, and our progress in drafting the results and analysis. Post which we delivered an oral presentation of findings and observations and then final documenting the finalised polished project report for submission.

## X. Deliverables

### A. A working system that predicts Ethereum next days prices using GPT-4o LLM server using API calls based on historical blockchain dataset which should be resisting all the external manipulations.

### B. A final report documenting re-defined problem statement, proposed solutions, techniques used, evaluation and updated related works and all the references used.

### C. A final presentation showcasing completetion of acheived target as per defined in the project outlines.

### D. Adding all the jupyter notebook and datasets used into github link provided in the report in Appendix.

## XI. Conclusion

We have come to value the volatility of crypto markets and the complexities in predicting Ethereum prices throughout this project. From studying traditional market simulation methods and development to GPT-4-enabled manipulation-proof predictions, we understood that traditional models in a highly speculative market are very restrictive. Our final deployment not only demonstrated the viability of large language models in finance forecasting but also emphasized the value of on-chain features such as gas costs and transaction metrics. We validated the model's effectiveness with Mean Absolute Error (MAE) and observed how the inclusion of blockchain analytics improved prediction accuracy, making the decision to include real-world transaction data worthwhile. The project was considered successful as it met its goal of developing a forecasting pipeline that could handle manipulation-susceptible data environments with empirically measurable outputs. In the future, research can involve incorporating more detailed on-chain data such as wallet transactions, miner behavior, and NFT-based transactions. We also envision incorporating real-time data streams along with fine-tuning language models for domain-specific accuracy. Additional improvements can be ensemble learning models that blend LLMs with traditional forecasting methods, offering robustness across diverse market conditions. Finally, this research lays the groundwork for scalable, intelligent crypto prediction tools applicable to academic and financial industry applications both.

Going forward, we would also like to incorporate a self-developed high-level model able to identify patterns of manipulation within datasets and purge them before making predictions, creating more prediction integrity. This could be achieved via anomaly detection models like Isolation Forest or Autoencoders trained on past data without manipulation. Applying unsupervised learning in order to discover outlier behavior and then pairing it with LLMs could lead to a hybrid solution for manipulation-proof forecasting. This add-on would introduce interpretability and stability to forecasted outcomes within a highly fluctuating crypto market. Lastly, this work offers a foundation for massive-scale, smart crypto prediction models both for pedagogy and utilization by the financial industry.

## XII. Appendix

**Click here to open the project presentation (PDF)**

The project is primarily build on Python 3.8.0. The integration takes place for the OpenAI API calls for GPT-4o and the total integration takes places as a matter of around 400 lines of code including actual works and related round work with different approaches to work ok.

- Pandas for dataframes
- scikit-learn for scaling and mean absolute error calculation
- matplotlib for plotting the predictions
- OpenAI API calls to make request calls for predictions, and it's a paid version around 10 minimum. It takes only 0.01 for a single call with occupying 50 tokens each call for the current provided dataset.
- Jupyter notebook to carry out the coding tasks in the ipynb files.

## REFERENCES


[1] Shiller, R. J. (2021). Irrational Exuberance: The Stock Market and Its Bubbles.
[2] Suri, P. (2022). Manipulation in Prediction Markets: Analysis and Solutions. arXiv:2201.08001.
[3] Li, Y., & Chen, Z. (2024). Machine Learning in Blockchain and Cryptocurrency: A Review. Journal of Financial Technology.
[4] Nakamoto, S. (2008). Bitcoin: A Peer-to-Peer Electronic Cash System. Retrieved from https://bitcoin.org/bitcoin.pdf
[5] Buterin, V. (2014). A next-generation smart contract and decentralized application platform. Ethereum White Paper. Retrieved from https://ethereum.org/en/whitepaper/
[6] Chen, Y., Xu, H., Ye, D. (2020). An overview of blockchain technology: Architecture, consensus, and future trends. Proceedings of the IEEE, 108(10), 1839–1870.
[7] Zhang, R., Xue, R., Liu, L. (2022). Security and privacy on blockchain. ACM Computing Surveys (CSUR), 54(8), 1-36.
[8] Gao, C., Li, Y., Wang, Z. (2023). ETHGasNet: A gas prediction model for Ethereum using deep learning. Journal of Blockchain Research, 5(1), 22-35.
[9] Jain, A., Choudhury, S. (2021). Cryptocurrency Price Forecasting using Social Media Sentiment Analysis. In 2021 IEEE International Conference on Big Data (Big Data), 4661-4669.
[10] Kim, H., Lee, M. (2023). Blockchain-based financial forecasting using machine learning and on-chain analytics. Finance Informatics Journal, 6(2), 77–91.


Priya Roy Master of Science In Computer Science, Illinois Institute of Technology, E-Mail: proy7@hawk.iit.edu.