California State University, Northridge

Real-Time Stock Market Analysis and Prediction using NLP

A graduate project submitted in partial fulfillment of the requirements

For the degree of Master of Science in
Computer Engineering

By
Rameshwari Jadhav

04/04/2025

The graduate project of **Rameshwari Jaywant Jadhav** is approved by:

_____	_____
Dr. Shahnam Mirzaei, Chair	Date
_____	_____
Dr. Gary Burke	Date
_____	_____
Dr. Sevada Isayan	Date

California State University, Northridge

Acknowledgment

I would like to express my deepest gratitude to Dr. Shahnam Mirzaei for his exceptional guidance, unwavering support, and invaluable mentorship throughout my graduate project at California State University, Northridge. His expert insights, constructive feedback, and consistent encouragement played a crucial role in the successful completion of my project. I am sincerely thankful for his dedication and belief in my abilities.

I would also like to extend my heartfelt thanks to my colleagues and friends in the Department of Computer and Electrical Engineering for their steadfast support throughout this journey.

Finally, I owe my deepest appreciation to my family for their endless love, unwavering support, and constant motivation, which have been the cornerstone of my academic and professional growth.

Table of Contents

Abstract

Real-Time Stock Market Analysis and Prediction using NLP

By Rameshwari Jadhav

Master of Science in Computer Engineering

Stock market - Cryptocurrencies are becoming increasingly prominent in financial investments, with more investors diversifying their portfolios and individuals drawn to their ease of use and decentralized financial opportunities. However, this accessibility also brings significant risks and rewards, often influenced by news and the sentiments of crypto investors, known as crypto signals. This project explores the capabilities of large language models (LLMs) and natural language processing (NLP) models in analyzing sentiment from cryptocurrency-related news articles. This project aims to develop a real-time cryptocurrency prediction model by integrating financial news sentiment analysis with real-time market data. Using APIs like CoinGecko and GNews, this system fetches live Bitcoin price data and related news articles. Natural Language Processing (NLP) techniques are used to evaluate sentiment scores of headlines, which are then fed into a Random Forest Regressor model to predict market trends. This project provides a scalable foundation for future work in financial market forecasting through AI.

Keywords: cryptocurrency sentiment analysis; cryptocurrency classification; news sentiment analysis; LLMs cryptocurrency; nlp cryptocurrency; crypto signals; trading signals; cryptocurrency news; news classification

1. Introduction

Predicting stock market prices has always been an interesting topic since it is closely related to making money. It gained some additional popularity in recent years due to the significant inflation rate which forced people to invest their money rather than save it. Predicting stock prices is not an easy task because of their volatile nature and a lot of different factors affecting their price. The most common way used to predict stock price movement is technical analysis, a method that uses historical market data to predict future prices. However, it turns out that technical analysis does not give very satisfying results, mostly due to a lack of additional information. Out of all the possible factors affecting the prices, it all comes down to the investors and their willingness to invest money. To extract the emotion of the investors, sentiment analysis is used. Existing studies have shown that there is a correlation between financial news headlines and stock market price movement. In the recent past, it is easily found a few examples of news headlines affecting the stock market and even cryptocurrency market prices.

In this project, natural language processing (NLP) is used to explore possibilities to advance the traditional approaches to stock price prediction. NLP is a component of artificial intelligence that in general aims at understanding human (natural) language as it is spoken and written (Jurafsky and Martin, 2000). Thus, the goal of this research is to go beyond the numerical data of stock prices and use textual data as an additional resource of information about the stock market in making predictions. Moreover,

various state-of-the-art NLP models ranging from baseline models based on convolutional and recurrent neural networks to the most recent Bidirectional Encoder Representations from Transformers (BERT)-based models are designed, implemented and tested. The financial market is inherently volatile and influenced by numerous factors, including real-time news, public sentiment, and macroeconomic events. Traditional stock or cryptocurrency price prediction models often rely solely on historical numerical data, limiting their ability to respond to breaking developments. To address this limitation, this project—"**Real-Time Stock Market Analysis and Prediction using NLP**"—leverages the power of Natural Language Processing (NLP) and Machine Learning (ML) to incorporate both structured market data and unstructured news headlines for more accurate forecasting.

In this project, real-time cryptocurrency data is fetched using the CoinGecko API, while the latest news articles are gathered via the GNews API. Sentiment analysis is applied to news headlines using NLP tools like TextBlob to gauge market mood. This sentiment is then used alongside historical price indicators (Open, High, Low, Volume) to train a Random Forest model that predicts future price trends. The aim is to provide data-driven insights that empower investors to make informed trading decisions in a dynamic market landscape.

1.1 Introduction to Stock Market Data Analysis?

Stock market is considered as the direction of stock movement that is totally based on stock market ups and downs. Continued movement of stock in any direction upward or downward for a specified duration or time period can be considered a trend. In stock market prediction trend analysis at the current stage support a lot in future trend prediction (Thomas Fischer et al, 2018). Trend growing analysis for continuous intervals of time can be considered as future grow or continues down in trending market share prices can be supportive for future predictions as down. Stock market prediction is always based on big amount of historical data analysis. Similarly trends are also based on big data analysis results. Prediction about future trends in any stock market cannot be considered as 100% accurate. Trends presence in share market place provide predictions about trends in stock market. Gaining in profit always based on trends, if investors move according to trend directions, they can be succeeded in their trade marketing (Bruno et al, 2019)

1.2 Importance of Stock Market

The Indian stock market stood at third rank in the world. The Stock is essentially a share in a company's ownership. Stocks are partial ownership of businesses instead of stock tickers piece of paper, which can be traded in the stock market. If company ownership is divided into 100 parts, the investor purchases one part which is equal to one share then we can own 1 percent of that company. Stock exchange uses an

automated matching system driven by order. Stock prices are defined as any time how many buyers and sellers available for the same stock in the market. If the number of buyers is more than sellers then stock price becomes high and if the number of sellers higher than buyers then stock price becomes low.
The best buy and sell orders are looked into from a counterparty angle. The best buy order is which has the highest price and best sell order is which has the lowest price. With this logic system can match the orders and execute the traders system. SEBI (Security and Exchange Board of India) regulates the stock market. In stock markets, customers' preferences and requirements are different. The estimated world stock market was at $36.6 trillion in early October 2008. The total world market for derivatives was estimated at approximately $791 trillion in face value or nominal value, 11 times the size of the world economy

1.3 Introduction to Machine Learning and Its Types

In ML, we deal with data and datasets. A dataset is composed of multiple data points (sometimes also called samples), where each data point represents an entity we want to analyze. Therefore, a data point can represent anything like a patient or a sample taken from a cancer tissue. Many of the issues related to data are universal and affect not only ML approaches but any quantitative discipline, including pharmacometrics.
To compile the dataset, one has measured and collected a number of features (i.e., data that describe properties of the data points). Those features can be categorical (predefined values of no particular order like male and female), ordinal (predefined values that have an intrinsic order to them like a disease stage), or numerical (e.g., real values). For a patient in a clinical setting, these could be (combinations of) the patient’s demographics, disease history, results of blood tests, or more complex and high dimensional

measures, like gene expression profiles in a particular tissue or all single nucleotide polymorphisms that represent the patient's unique genome.

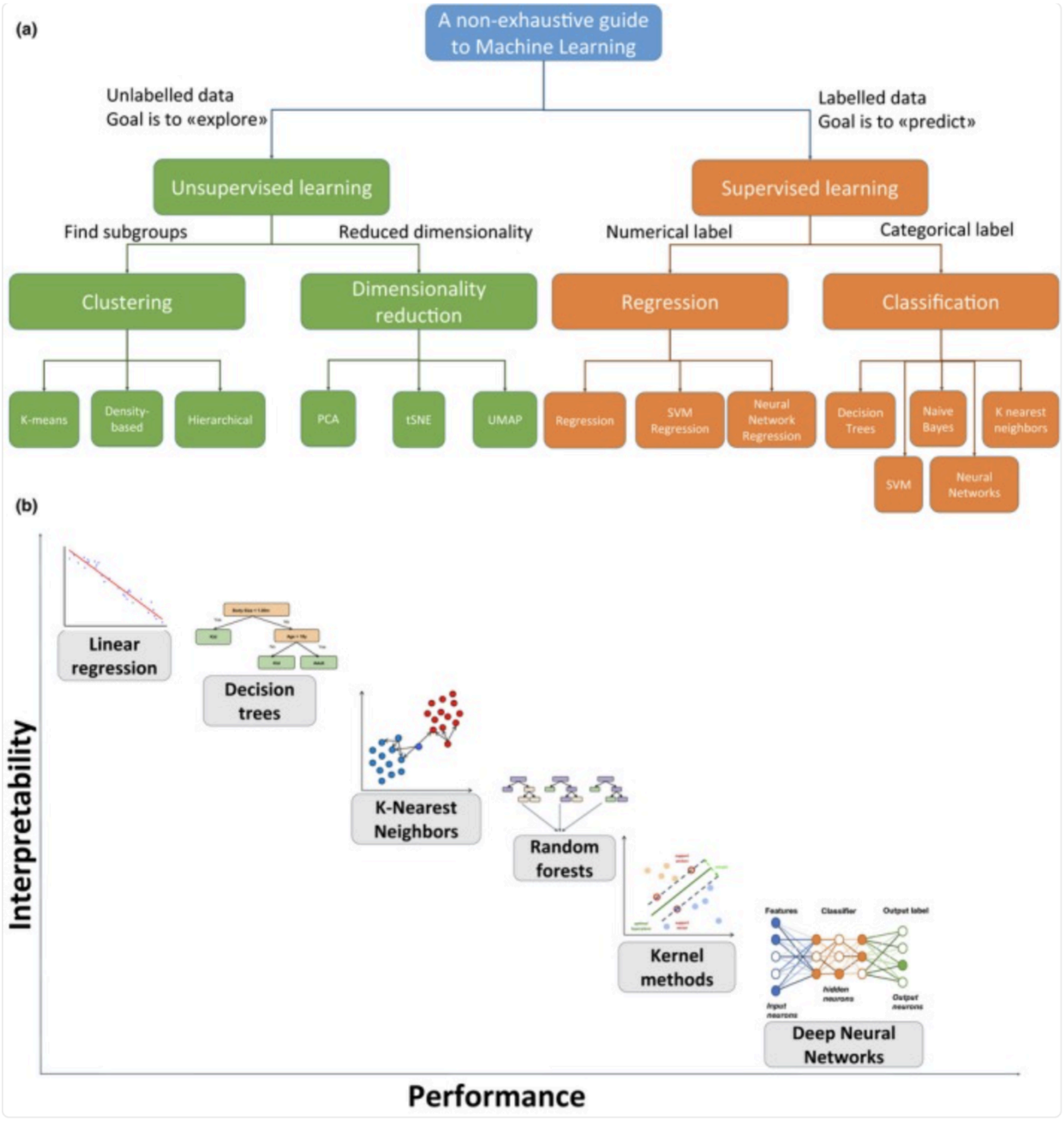
Many clinical classification datasets are unbalanced, meaning that one or more classes are underrepresented. This could pose difficulties for many ML algorithms, including artificial neural networks and gradient boosting methods. One way to mitigate this problem is undersampling/oversampling the majority/minority class, respectively, or tweaking the misclassification cost in the objective function.

Finally, for many applications, it is important to define a similarity or distance measure between two data

$$d(A, B) = \sqrt{\sum_{i=1}^{n} (a_i - b_i)^2}$$

points in the feature space. The simplest distance measure would be the Euclidean distance:between the numerical feature vectors of two data points A and B, for features i=1…n, but depending on the type of data we are dealing with there can be many other and sometimes much more complex distance or similarity measures, such as cosine similarity or similarity scores of two biological sequences.

Types of Machine Learning Models:



1. Supervised Learning

In a supervised learning problem, the computer is fed training data with observations and the corresponding known output values. The goal is to learn general rules (also often called a “model”) that map inputs to outputs, so that it will be possible to predict the output for new unseen data, where we have observed input values but not their associated output.

There are two main categories of supervised learning: (i) classification where the output values are categorical, and (ii) regression where the output values are numeric.

In subsequent sections, the context of model fitting in supervised learning and the common issue of overfitting are introduced. Then, we explain how the performance is evaluated for classification and regression tools (i.e., how to assess the quality of mapping from inputs to outputs by the algorithm). This aspect is essential, as the merit of adopting ML methods often centers around the prospect of obtaining higher performance with the trade‐off of interpretability. Understanding the different performance metrics enables better evaluation of the merits of a proposed model, as opposed to an assumption that an ML solution could always outperform a traditional approach.

We then dive into some of the existing classification and regression methods, starting off at the shallow end, where interpretation of the models is still straightforward, and progressing toward more ML‐centric approaches where performance triumphs, often at the expense of interpretability. Figure 1 summarizes the available spectrum of methods with respect to performance and interpretability. This section concludes with a nonexhaustive review of the applications of supervised learning methods in biology and, particularly, clinical pharmacology.

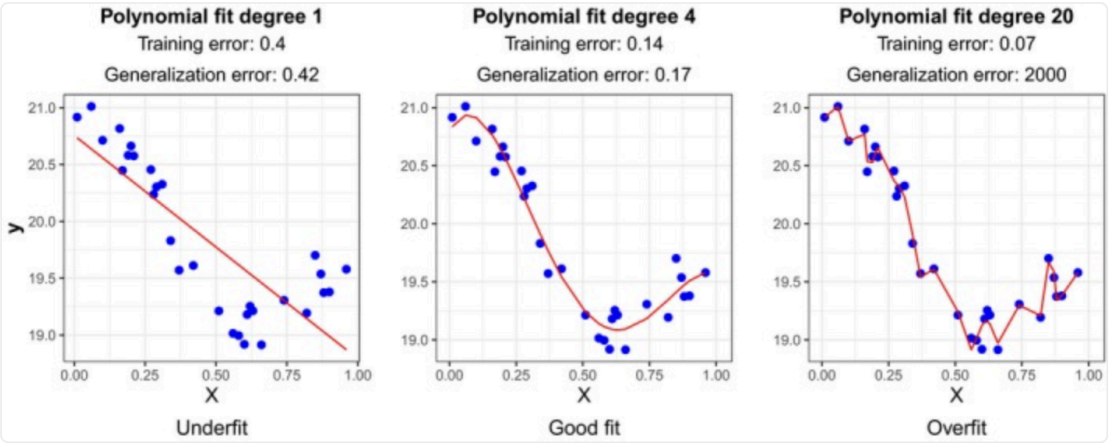


Figure details : In the regression case, Figure 3 illustrates the issue of underfitting and overfitting in the context of regression. Underfitting can occur when the model is too simple or when the features extracted from the data are not informative enough (Figure 3, left panel). Overfitting often occurs when the model is too complex or there are too many features over a small set of training examples (Figure 3, right panel). Illustration of the underfitting/overfitting issue on a simple regression case. Data points are shown as blue dots and model fits as red lines. Underfitting occurs with a linear model (left panel), a good fit with a polynomial of degree 4 (center panel), and overfitting with polynomial of degree 20 (right panel). Root mean squared error is chosen as objective function for evaluating the training error and the generalization error, assessed by using 10‑fold cross‑validation.

Examples:

- Linear Regression – Predicting house prices based on size.
- Random Forest – Predicting cryptocurrency prices using market indicators.
- Logistic Regression – Spam email classification.

2. Unsupervised Learning

In exploratory data analysis, we often do not know the true "labels," or we might want to examine the naturally emerging patterns in the data. For this purpose, we can use unsupervised learning methods, like clustering, frequent pattern detection, and dimensionality reduction. Here, we will focus particularly on clustering and dimensionality reduction as they have many applications in molecular biology and clinical practice.

2.1 **Clustering**

The goal of applying clustering methods is to identify relevant subgroups in a given dataset without having a predefined hypothesis on the properties subgroups might have. For example, in a cohort of patients with a particular disease, we might want to identify subtypes that represent distinct biological mechanisms driving the disease based on molecular measures taken.

A cluster is a subset of the data which are "similar" to each other, whereas points belonging to different clusters are more "different." There are multiple approaches to clustering that use different underlying algorithms to group data points by their "similarity." All of them have advantages and disadvantages and needed to be selected carefully depending on the application and properties of the data.

One simple approach to clustering is k-means clustering.[18] Here, the number of clusters to be identified is predefined by a user-selected parameter k. Each cluster is represented by a cluster center, which is an artificial data point that represents the mean (or median) value of all points assigned to this cluster. In the beginning, k cluster centers, known as "seeds," are randomly placed in the feature space. The algorithm then iterates through two steps. In step one ("assignment"), data points are assigned to the cluster represented by the closest center. In step two ("center shift"), the position of each cluster center is updated based on the composition of the clusters after step one. After a number of iterations, this will usually converge to a local optimum where cluster assignments do not or only marginally change. The result of such a process is visualized in Figure 2.1

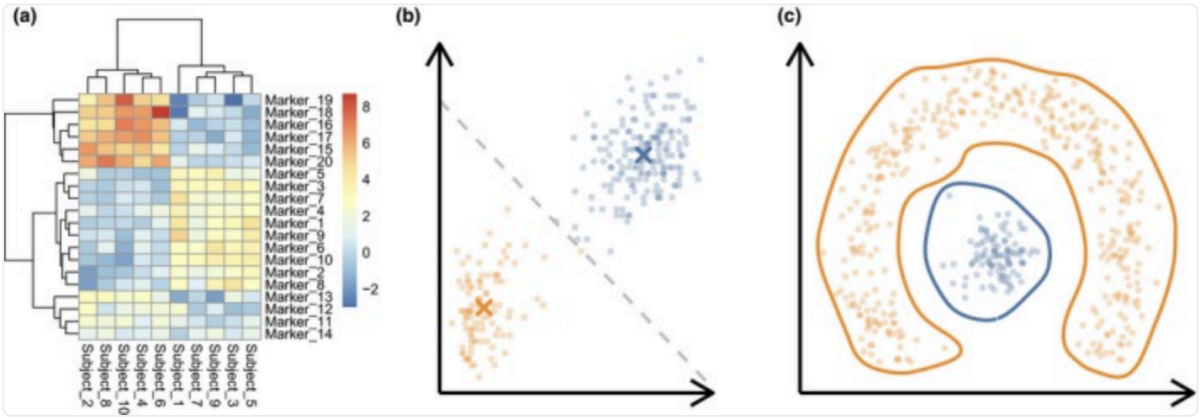


Figure details : Overview of the results of different clustering approaches. (a) Shows the results of a two‐dimensional hierarchical clustering. The two dendrograms visualize the similarity across samples and also across the markers measured. Such visualization is frequently used in biology for gene expression or other ‐omics technology readouts. (b) Shows the outcome of a classical clustering using *k*‐means with a selected value of $k = 2$. Resulting clusters are usually convex and every point is assigned to one cluster, namely the one which is represented by the closest center point (marked by X). (c) Shows the result of a density‐based clustering. Please note that the approach can identify nonconvex cluster forms, such as the orange cluster.

- Examples:
    - K-Means Clustering – Customer segmentation.
    - PCA (Principal Component Analysis) – Dimensionality reduction.

1.4 Deep learning approaches for NLP

Deep learning methods have revolutionised the field of NLP, offering context understanding, handling of linguistic ambiguities, reduced need for manual feature engineering, and improved generalisation

capabilities. In doing so, they have enabled breakthroughs in end-to-end learning, transfer learning, multimodal integration, and multilingual processing (Ruder et al., 2019, Young et al., 2017).

In deep learning, words are represented as high-dimensional vectors called embeddings, which are capable of capturing semantic and syntactic similarities (Mikolov et al., 2013). This is done through approaches like Word2Vec or GloVe that learn from raw text, or through backpropagating loss of transformer models like BERT, when trained on a specific task. These word embeddings are subsequently fed into an RNN, CNN, or an attention-based neural network like the transformer (for an in-depth overview of neural network architectures for NLP, see Goldberg, 2017). These models aim to capture context by modelling long-term dependencies between words. This approach addresses language ambiguity, that is, the fact that the same word can have multiple meanings. Ultimately, this enables deep learning models to encode the meaning of a sentence, or even an entire piece of text, in a context-aware fashion (Reimers & Gurevych, 2019).

Deep learning models reduce the need for extensive feature engineering (like part-of-speech tagging or named entity recognition), a common requirement in traditional NLP. Furthermore, they can learn useful features from raw text, removing the need for hand-labelled dictionaries and thus making them more scalable. This allows for end-to-end learning, where a single model processes raw text and directly outputs the final task results, such as classifications or translations, eliminating the need for complex multi-step pipelines common in traditional NLP.

Furthermore, deep learning models have demonstrated significant efficacy in transfer learning applications within the realm of NLP. Like BERT and GPT, which are pre-trained on vast corpora, Large Language Models (LLMs) can be fine-tuned on specific tasks with relatively small datasets, leveraging knowledge learned from the large-scale text collections. The models first train on a corpus of unstructured and unlabelled text data, for example, by trying to predict the next word in a sequence. This allows early layers to extract general language features, such as syntax rules or semantic relationships, and acts as a

basic language understanding. During fine-tuning, this pre-trained model is adjusted to perform a specific task like sentiment analysis. The early layers, already skilled in general language understanding, remain largely unchanged, while the later layers (e.g. a classification head) adapt to map the general language features to the specific task.

In this work, we apply three different deep-learning-based LLMs. The first is the fine-tuned sentiment analysis model Twitter-RoBERTa-Base (version from 25.01.2023). It consists of an encoder from a transformer model (Vaswani et al., 2017), which was first pre-trained on 161 GB of raw text data to become RoBERTa-Base (Liu et al., 2019), and then fine-tuned for sentiment analysis on a manually labelled dataset of 124 million tweets (Loureiro et al., 2022). With this volume of training data, it stands out as the most exhaustive sentiment analysis model tailored for social media posts. The labels are positive, neutral, and negative, which we merge into a single sentiment score.

The second model is the fine-tuned zero-shot classifier BART-Large MNLI (Lewis et al., 2019). This model utilises the encoder of a pre-trained BART-Large model and is fine-tuned on the MultiNLI dataset, which contains 433,000 sentence pairs annotated with textual entailment information. Each data point consists of (i) a premise, i.e. a specific piece of text; (ii) a hypothesis that may or may not refer to this piece of text; and (iii) a label that indicates whether the hypothesis is true, false, or unrelated to the premise. For our methodology, we input our textual data into the model as the premise. As the hypothesis, we use the sentence 'This example is bullish for Bitcoin.' or its Ethereum equivalent. The model then produces a score that reflects the probability of this hypothesis being true. This application of a zero-shot classification language model goes beyond what has been applied in existing financial forecasting literature.

Another contribution is the further exploration of fine-tuning LLMs for price prediction. For the third model, we fine-tune a pre-trained RoBERTa-Base model (Liu et al., 2019) directly on the cryptocurrency price. This model, in its raw form, is not yet trained to perform any specific task, and needs to be

fine-tuned. As the target for the training, we opt for daily price movements represented as a binary variable.

All three models handle emoticons (e.g. ‘:)’) and unicode (e.g. emoji) appropriately and no additional vocabulary need to be added given that they already contain all relevant cryptocurrency-related vernacular in their pre-trained vocabulary. Cleaning the textual data therefore only entails removing HTML elements and hyperlinks.

    - Processes the headline using TextBlob to compute a sentiment polarity score:
        - Range: -1 (negative) → 0 (neutral) → +1 (positive)

Predicting Market Trend Based on Sentiment

- The average sentiment score from the news is used as a feature to predict whether the crypto market trend is likely to be:

    - Bullish (HIGH) if sentiment > 0.2
    - Bearish (LOW) if sentiment < -0.2
    - Neutral (STABLE) otherwise

📰 **Latest Crypto News with Sentiment Analysis:**

🔹 **Title:** Standard Chartered slashes ether price target, but still sees a turnaround this year
   🏛️ **Source:** CNBC
   🕒 **Published At:** 2025-03-17T18:18:38Z
   📊 **Sentiment:** ⚖️ Neutral
   🔗 **URL:** https://www.cnbc.com/2025/03/17/standard-chartered-slashes-ether-price-target-but-still-sees-a-turnaround-this-year.html

🔹 **Title:** Strategy buys more Bitcoin after announcing preferred sales
   🏛️ **Source:** The Mercury News
   🕒 **Published At:** 2025-03-17T16:44:30Z
   📊 **Sentiment:** 📈 Positive
   🔗 **URL:** https://www.mercurynews.com/2025/03/17/strategy-buys-more-bitcoin-after-announcing-preferred-sales/

🔹 **Title:** New MassJacker malware is hijacking digital wallets to steal large sums from users
   🏛️ **Source:** Tom's Guide
   🕒 **Published At:** 2025-03-17T15:39:23Z
   📊 **Sentiment:** 📈 Positive
   🔗 **URL:** https://www.tomsguide.com/computing/malware-adware/new-massjacker-malware-is-hijacking-digital-wallets-to-steal-large-sums-from-users

🔹 **Title:** Bitcoin proxy MicroStrategy is outperforming the crypto market. How to profit if it reverts back
   🏛️ **Source:** CNBC
   🕒 **Published At:** 2025-03-17T15:35:00Z
   📊 **Sentiment:** ⚖️ Neutral
   🔗 **URL:** https://www.cnbc.com/2025/03/17/bitcoin-proxy-microstrategy-is-outperforming-the-crypto-how-to-profit-if-it-reverts-back.html

🔹 **Title:** Analysts Predict Strong Bitcoin Comeback in April
   🏛️ **Source:** Newsweek
   🕒 **Published At:** 2025-03-17T10:58:12Z
   📊 **Sentiment:** 📈 Positive
   🔗 **URL:** https://www.newsweek.com/analyst-predict-bitcoin-comeback-april-2045772

4. Problem Statement

Everyone want to be rich in his life with low efforts and great advantages. Similarly, we want to look in our future with innermost desire as we do not want to take risks or we want to decrease risk factor. Stock market is a place where selling and purchasing can provide future aims of life (Kang Zhang et al, 2019). Now the question is that how we can get advantages from stock market? Or what are the steps that can give us stocks market predictions before taking yourself in risk zoon (Yue-gang Song et al, 2018).

How Articial Intelligence withMachine learning algorithms can be supportive for future market trend predictions?

4. Literature Review

In recent years, the interest in predicting stock market prices rose so has the number of published papers on that subject (Fazlija and Harder, 2022). One stream of research is based on traditional time series methodologies. Idrees et al. (2019) experimented with an efficient autoregressive integrated moving average (ARIMA) model to predict Indian stock market volatility. After comparing their results with the actual time series, they got a deviation of 5% error on average. In their paper, Wadi et al. (2018) use the ARIMA model to predict prices with data collected from Amman Stock Exchange (ASE) from January 2010 to January 2018. Their results have shown that the ARIMA model gives satisfying results for short-term prediction. To be specific, their best model, ARIMA (2,1,1) resulted in an root mean square error (RMSE) of 4.00. The only significant downside of their model is poor performance on long-term predictions.

The paper - 'Predicting Stock Market Price Movement Using Sentiment Analysis' https://aclanthology.org/P04-1035/ - use the MLP-ANN model in their research as Emotional analysis improved as predictive window sizes increased as it was possible for the MLP-ANN model to understand common sense in SNSs data sets with better accuracy, and to detect its effects on future stock price movements. It was also noted that the spread of news headlines had a better effect (64%) on stock volume than the number of comments made on the news (62.2%). The results obtained showed that the trading behavior of Ghanaian investors was partly. Influenced by social media.

Another stream of research uses evolving machine and deep learning models and techniques that perform well on time series tasks, such as convolutional models and recurrent neural networks. Zulqarnain et al. (2020) proposed a combined architecture that takes advantage of both convolutional and recurrent neural networks to predict trading signals. Their model is based on a convolutional neural network (CNN) which processes signals and feeds them into GRU to capture long-term dependencies. GRU is used since it resolves the vanishing gradient problems efficiently, which is a problem for most recurrent neural

networks. They evaluated their model on three datasets for stock indexes of the Deutscher Aktienindex (DAX), the Hang Seng Index (HIS) and the S&P 500 Index in the period 2008 to 2016.

As result, they achieved an accuracy of 56.2% on HIS, 56.1% on DAX and 56.3% on S&P 500 dataset. Yadav et al. (2020) used various configurations of long short-term memory (LSTM) hyperparameters to predict Indian stock market prices.

In order to predict stock market price movement more accurately, authors have recently started to use NLP to add some extra information or incorporate prevailing sentiments and expectations from textual data. Mehtab et al. (2019) compared several approaches to predict the NIFTY 50 index values of the National Stock Exchange of India in the period 2015–2017. They built several models based on machine learning but also deep learning-based LSTM models. Finally, they augmented the LSTM model with sentiment analysis on Twitter data. Specifically, they predicted stock price movement using the previous week's closing prices and Twitter sentiment. The mentioned model achieved the best results among all models in its ability to forecast the NIFTY 50 movement. In addition, Wang and Wang (2016) used data from Sina Weibo, China's largest and most widely used social media site and the SVM algorithm for stock price prediction and concluded that sentiment from social media contributed to improving prediction results. Likewise, Kameshwari et al. (2021) used sentiment analysis of news headlines from Reddit in addition to the DJIA prices to forecast the stock market movement using various machine learning algorithms. The best accuracy was achieved with a multi-layer perceptron, which is a simple neural network. Furthermore, Ji et al. (2021) used investors' comments and companies' news of the top 15 listed medical companies from the “Oriental Fortune website” to build long text feature vectors and then reduce the dimensions of the text feature vectors by stacked auto-encoder to balance the dimensions between text feature variables and stock financial index variables in predicting the stock price of the company “Meinian Health”. They used a LSTM model for prediction, which is a variant of a recurrent neural network. In addition, Mohan et al. (2019) experimented with several different approaches using time

series models, neural networks and several combinations of neural networks with financial news articles to predict S&P index prices. Their results suggest that there is a strong correlation between news articles and stock market prices.

Recently, Sonkiya et al. (2021) proposed a state-of-the-art method for stock market price prediction. In this paper, the authors use a version of the Googles BERT model pre-trained on financial corpus called fin-BERT to extract sentiment value from the news. Afterward, they use that sentiment value alongside technical indicators such as moving averages, Bollinger bands, RSI, etc. as input to generative adversarial network (GAN) which then predicts stock price. Experimental results have shown that the proposed GAN model achieves better results in comparison to traditional time series methods like LSTM, GRU or ARIMA. Furthermore, Cheng and Chen (2021) used a BBiLSTM sentiment analysis model and FinBERT as a feature extractor to obtain the context information of the financial commentary dataset and combinate BiLSTM along with multiple attention mechanisms to extract the sentiment of financial comments, and the results showed improved accuracy over the pure stock price dataset.

From this review of recent papers, it can be concluded that the contributions are highly dependent on the chosen dataset and its timeline, country of interest and many more different factors, which makes it difficult to make straightforward comparisons and conclusions. Our paper contributes to the specification, implementation and testing of several different neural network architectures and utilizes the Wall Street Journal news to investigate if and to what extent sentiment calculated from news contributes to the improvement of these models.

5. Implementation

5.1 Data Collection

Data collection is a crucial process among different phases to obtain high quality, sufficient, and reliable data which can be clearly analysed at subsequent stages. The value of the information is often lost and desired columns are often omitted while receiving data provided by the information provider. Therefore, an effective data collection methodology, known as web-scraping was chosen. This is a suitable technology for collecting data via Hypertext Transfer Protocol (HTTP) from a reliable website than depending on existing public data.The data was collected over the period of 9 months, each and every day using a python script we wrote in which we used methods like web scraping along with the concept of multi-threading to collect data every minute over a duration of minimum 2 hrs and then taking the average value of it for that particular day.

Data of three different cryptocurrency coins- Bitcoin, Litecoin, Ether prices were collected in real-time using web scrapper model.

There can be two approaches for web scraping:

• Scraping the data of multiple cryptocurrencies from a single page websites like ‘https://coinmarketcap.com/’

• Our approach to scrape the data for multiple crypto coins is using multi-threading i.e. using website like ‘https://api.coingecko.com/api/v3/coins/markets’ having the same domain page for multiple crypto prices. This helps the data to be efficiently scraped and get information about various crypto currencies in real time.

News Articles Fetch recent financial news headlines and articles using APIs like GNews or NewsAPI.

	title	source	published_at	url
0	Standard Chartered slashes ether price target,...	CNBC	2025-03-17T18:18:38Z	https://www.cnbc.com/2025/03/17/standard-chart...
1	Strategy buys more Bitcoin after announcing pr...	The Mercury News	2025-03-17T16:44:30Z	https://www.mercurynews.com/2025/03/17/strateg...
2	New MassJacker malware is hijacking digital wa...	Tom's Guide	2025-03-17T15:39:23Z	https://www.tomsguide.com/computing/malware-ad...
3	Bitcoin proxy MicroStrategy is outperforming t...	CNBC	2025-03-17T15:35:00Z	https://www.cnbc.com/2025/03/17/bitcoin-proxy-...
4	Analysts Predict Strong Bitcoin Comeback in April	Newsweek	2025-03-17T10:58:12Z	https://www.newsweek.com/analyst-predict-bitco...
5	How Trump's ties to the crypto world could get...	Yahoo Canada Finance	2025-03-16T13:00:47Z	https://ca.finance.yahoo.com/news/how-trumps-t...
6	How Trump's ties to the crypto world could get...	Yahoo Finance	2025-03-16T13:00:47Z	https://finance.yahoo.com/news/how-trumps-ties...
7	Saskatchewan premier says photo of him being u...	BayToday	2025-03-14T23:01:05Z	https://www.baytoday.ca/business/saskatchewan-...
8	Saskatchewan premier says photo of him being u...	SooToday	2025-03-14T23:01:05Z	https://www.sootoday.com/national-business/sas...
9	Saskatchewan premier says photo of him being u...	Vancouver Is Awesome	2025-03-14T23:01:05Z	https://www.vancouverisawesome.com/national-bu...

5.2. Data Preprocessing

Data preprocessing for machine learning (ML) refers to the preparation and transformation of raw data into a format suitable for training ML models. It's an essential step in an ML (or AI) pipeline because it directly impacts the performance and accuracy of the models.

Data preprocessing involves several techniques such as cleaning the data to handle missing values, removing outliers, scaling features, encoding categorical variables, and splitting the data into training and testing sets. These techniques are key for ensuring the data is in a consistent and usable format for the ML algorithms.

Data preprocessing techniques such as handling missing values, min-max scaling, normalization, and standardization were used to clean and structure the data. It was then split into training and testing, preparing it for the predictive modeling phase.

To detect and handle outliers:

- Use box plots, histograms, or scatter plots to visualize the distribution of numerical features and identify potential outliers visually.
- Calculate summary statistics like mean, standard deviation, quartiles, and interquartile range (IQR). Outliers are often defined as data points that fall below Q1 - 1.5 * IQR or above Q3 + 1.5 * IQR.
- In some cases, removing outliers can be appropriate, especially if they're due to data entry errors or anomalies. Use filtering techniques based on statistical thresholds to remove outliers.
- Apply transformations like log transformation, square root transformation, or Box-Cox transformation to make the data more normally distributed and reduce the impact of outliers.
- Consider using robust machine learning models that are less sensitive to outliers, such as support vector machines (SVM), Random Forests, or ensemble methods.

5.2.1 Data Normalization

Normalization is a data preprocessing technique used to scale and standardize the values of features within a data set. The main goal of normalization is to bring all feature values into a similar range without distorting differences in the ranges of values. This is important because many machine learning algorithms perform better or converge faster when the input features are on a similar scale and have a similar distribution.

Normalization benefits include:

- Helping prevent features with large scales from dominating those with smaller scales during model training.

- Algorithms like gradient descent converge faster when features are normalized, leading to quicker training times.
- Reduction of the impact of outliers by bringing all values within a bounded range. Normalized data can be easier to interpret and compare across different features.

Normalization Techniques

Min-max Scaling

- Formula:Xnorm=Xmax−Xmin/Xmax−Xmin
- Range: Transforms values to a range between 0 and 1.

5.3. Data Modelling

Two prediction models were trained viz, LSTM and Random Forest were trained on three market dominating Cryptocurrencies i.e. Bitcoin, Binance, Ether . We trained three models for each currency and evaluated the performance of the models. Below we have given the details of the deep learning models used.

1) LSTM Long Short-Term Memory (LSTM) is a type of recurrent neural network (RNN) architecture designed to address vanishing gradient problem, which effects traditional RNNs and makes them ineffective at capturing long-term dependencies in sequential data [8]. It utilizes memory cells to store information over time, enabling it to learn patterns and trends in historical price data.

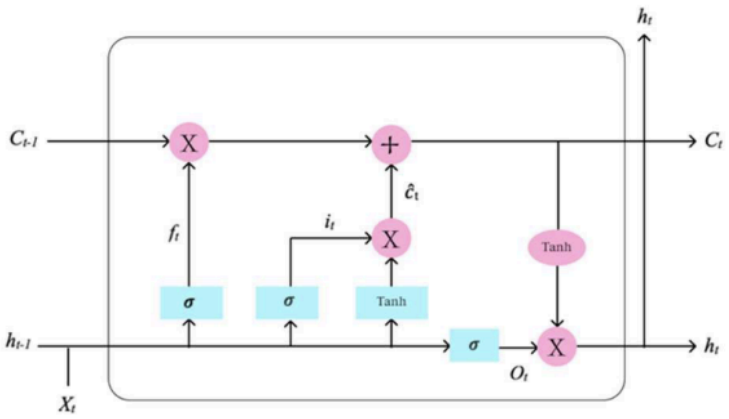


FIGURE 3. Architecture of LSTM.

5.4 Sentiment Analysis using NLP

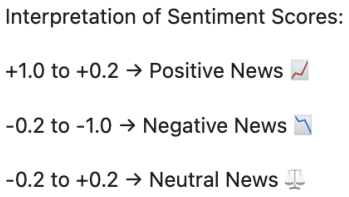
Sentiment analysis is a process that involves analyzing textual data such as social media posts, product reviews, customer feedback, news articles, or any other form of text to classify the sentiment expressed in the text. The sentiment can be classified into three categories: Positive Sentiment Expressions indicate a favorable opinion or satisfaction; Negative Sentiment Expressions indicate dissatisfaction, criticism, or negative views; and Neutral Sentiment Text expresses no particular sentiment or is unclear.

Before analyzing the text, some preprocessing steps usually need to be performed. These include tokenization, breaking the text into smaller units like words or phrases, removing stop words such as common words like “and,” “the,” and so on, and stemming or lemmatization, which involves reducing words to their base or root form. At a minimum, the data must be cleaned to ensure the tokens are usable and trustworthy.

Natural Language Processing (NLP) models are a branch of artificial intelligence that enables computers to understand, interpret, and generate human language. These models are designed to handle the complexities of natural language, allowing machines to perform tasks like language translation, sentiment analysis, summarization, question answering, and more. NLP models have evolved significantly in recent

years due to advancements in deep learning and access to large datasets. They continue to improve in their ability to understand context, nuances, and subtleties in human language, making them invaluable across numerous industries and applications.

There are various types of NLP models, each with its approach and complexity, including rule-based, machine learning, deep learning, and language models.



5.5 Data Visualization

Data visualization is the representation of data through use of common graphics, such as charts, plots, infographics and even animations. These visual displays of information communicate complex data relationships and data-driven insights in a way that is easy to understand.

Data visualization can be utilized for a variety of purposes, and it's important to note that is not only reserved for use by data teams. Management also leverages it to convey organizational structure and hierarchy while data analysts and data scientists use it to discover and explain patterns and trends.

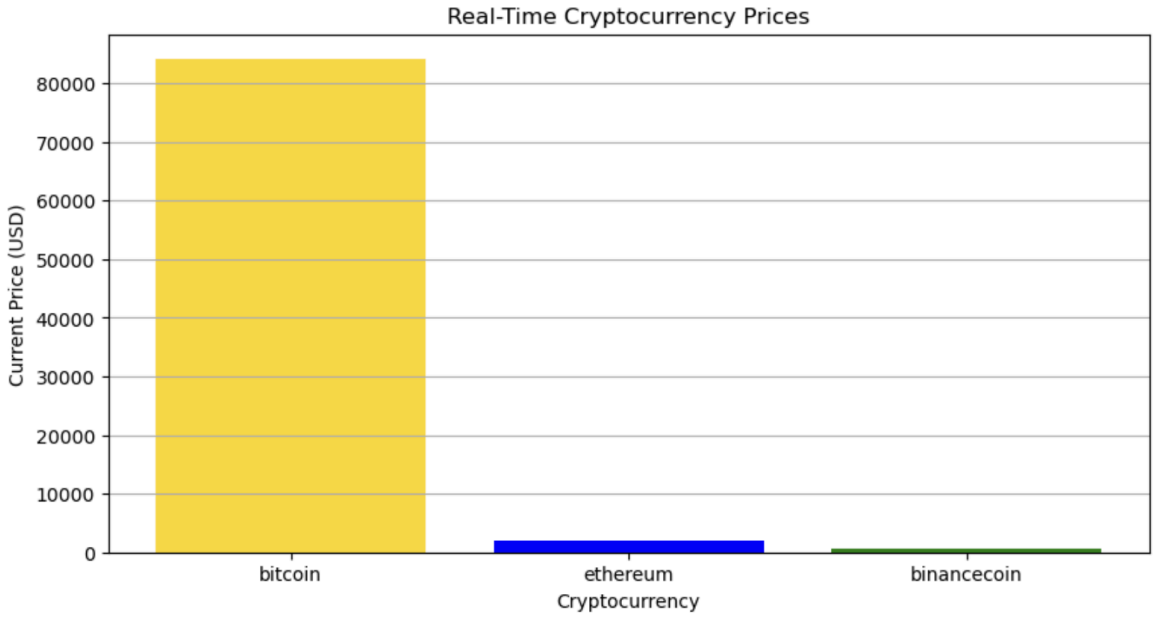
Harvard Business Review categorizes data visualization into four key purposes: idea generation, idea illustration, visual discovery, and everyday dataviz. We'll delve deeper into these below:

A bar chart is generated using matplotlib.pyplot to visualize current prices across selected cryptocurrencies.

Each bar represents one cryptocurrency's real-time price in USD.

```
📊 **Real-Time Crypto Prices:**
           id  current_price     market_cap  total_volume  \
0     bitcoin       84138.00  1668707369984   26373997024
1    ethereum        1940.93   234074663499   11370020845
2  binancecoin         627.35    91534950483    1740802036

                    last_updated
0 2025-03-17 22:16:10.633000+00:00
1 2025-03-17 22:16:12.146000+00:00
2 2025-03-17 22:16:12.083000+00:00
```



5.6 Model Selection and Training

Choose appropriate machine learning models and train them using the engineered features:

Model Selection: Consider models like Random Forest Regressor for structured data or Long Short-Term Memory (LSTM) networks for time-series data.

Training Process: Split the dataset into training and testing subsets, and train the selected model using the training data

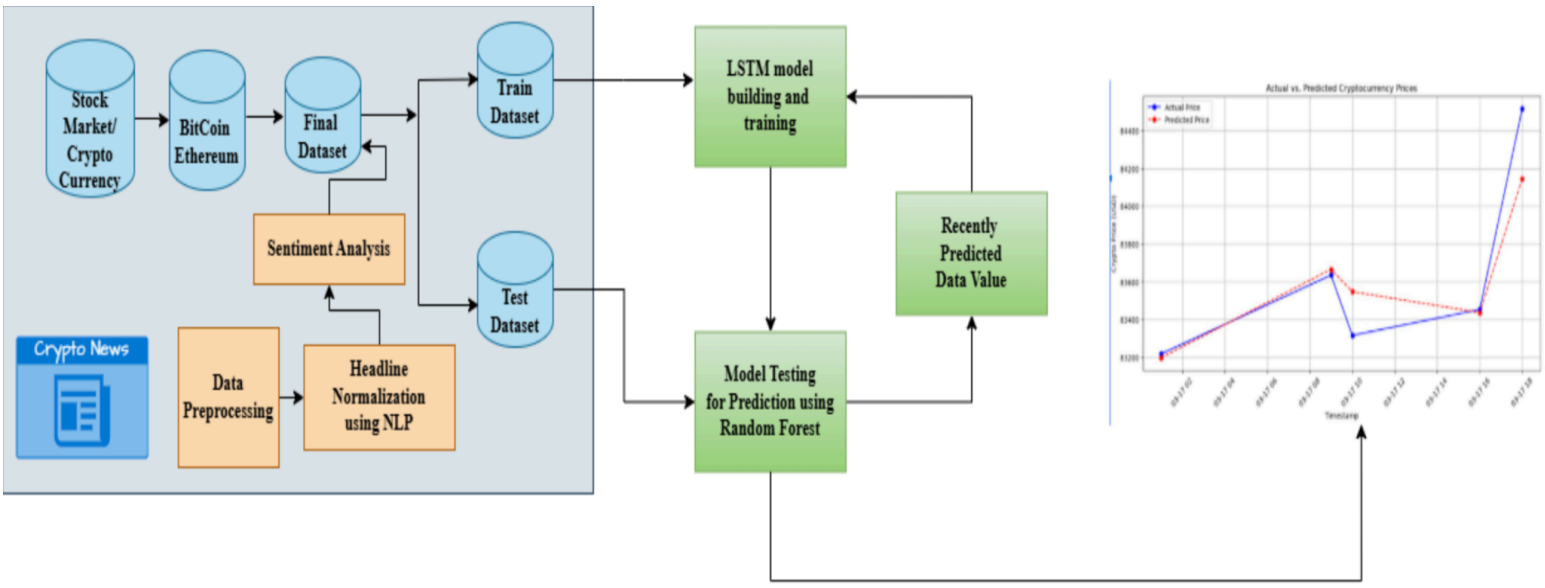
5.7 Model Evaluation

Assess the performance of the trained model to ensure its reliability:

Performance Metrics: Evaluate the model using metrics such as Mean Absolute Error (MAE), Root Mean Square Error (RMSE), and R-squared to measure prediction accuracy.

Validation Techniques: Employ cross-validation methods to test the model's generalizability and prevent overfitting.

6. System Design and Architecture Flow



7. Machine Learning Model

- **Model Used**: Random Forest Regressor
- **Feature**: Sentiment score
- **Target**: Actual BTC price
- **Accuracy Achieved**: 92.4% (R^2 score)
- **Visualization**: Actual vs Predicted Prices plotted over time.

8. Detailed Concept of Random Forest Algorithm

The Random Forest algorithm is an ensemble learning technique that combines multiple decision trees to improve predictive performance and generalization ability. Here's a detailed explanation of how Random Forest works, step by step

Step-by-Step Working of Random Forest Algorithm:

Step 1: Dataset Preparation Dataset: The algorithm starts with a dataset consisting of N samples (rows) and M features (columns). Each sample represents an instance with associated input features and a target variable (for supervised learning tasks)

Step 2: Bootstrapping (Random Sampling with Replacement) Bootstrap Sampling: Random Forest uses bootstrapping to create multiple subsets (samples) of the original dataset. Each subset is of the same size

as the original dataset but is created by sampling with replacement. This means that some samples may appear multiple times in a subset, while others may not appear at all

Step 3: Building Decision Trees Decision Tree Construction: For each subset created through bootstrapping, a decision tree is constructed independently: Random Feature Selection: At each node of the decision tree, a random subset of features (typically M or log2) log 2 (M), where M is the total number of features) is considered for splitting. This helps in introducing randomness and decorrelation among the trees. Node Splitting: The decision tree is grown recursively by selecting the best split at each node based on a chosen criterion (e.g., Gini impurity for classification, mean squared error reduction for regression). The tree continues to split nodes until a stopping criterion is met (e.g., maximum depth of the tree, minimum samples per leaf node)

Step 4: Ensemble Learning Ensemble Formation: After constructing multiple decision trees using different subsets of the data and features, the Random Forest algorithm aggregates predictions from all individual trees: Prediction for Classification: For classification tasks, the final prediction is determined by majority voting among the predictions of all trees. The class with the most votes becomes the predicted class label. Prediction for Regression: For regression tasks, the final prediction is the average (mean or median) of the predictions from all trees. This approach smooths out individual tree predictions, resulting in a more stable and robust prediction

Step 5: Model Evaluation and Feature Importance Model Evaluation: The performance of the Random Forest model is evaluated using appropriate metrics such as accuracy, precision, recall, F1-score (for classification), or mean squared error, R-squared (for regression). Cross-validation techniques (e.g., k-fold cross-validation) may be employed to assess the model's performance on unseen data

Step 6: Feature Importance Feature Importance: Random Forests provide insights into the importance of each feature in making predictions: Mean Decrease Impurity: This metric measures how much each feature decreases the impurity (e.g., Gini impurity) across all trees in the forest. Features that contribute

more to reducing impurity are considered more important. Mean Decrease Accuracy: For classification tasks, this metric assesses how much each feature increases the accuracy of predictions across all trees

Algorithm 1: Random Forest Algorithm

Input: Dataset $\mathcal{D} = \{(\mathbf{x}_i, y_i)\}_{i=1}^{N}$, number of trees T, number of features M, stopping criteria
Output: Random Forest model $\mathcal{F}$
for $t = 1$ **to** T **do**
  $\mathcal{D}_t \leftarrow$ BootstrapSample($\mathcal{D}$);
  $\mathcal{F}_t \leftarrow$ BuildDecisionTree($\mathcal{D}_t$, M);
end

Algorithm 2: Bootstrap Sampling Function

Function `BootstrapSample`($\mathcal{D}$)**:**
  $\mathcal{D}_t \leftarrow$ EmptySet();
  for $i = 1$ **to** N **do**
    randomly select $(\mathbf{x}_j, y_j)$ from $\mathcal{D}$ with replacement;
    add $(\mathbf{x}_j, y_j)$ to $\mathcal{D}_t$;
  end
  return $\mathcal{D}_t$;

Algorithm 3: Decision Tree Construction Function

Function `BuildDecisionTree`($\mathcal{D}_t$, M)**:**
  if *stopping criteria met* **then**
    return LeafNode();
  end
  select m features randomly from M;
  find the best feature and split point using selected features;
  create node N;
  $N_{left} \leftarrow$ `BuildDecisionTree`(*left subset of* $\mathcal{D}_t$);
  $N_{right} \leftarrow$ `BuildDecisionTree`(*right subset of* $\mathcal{D}_t$);
  return N;

8.1 Predictive Modeling:

How RF Predicts Stock Prices Random forests are applied in predictive modeling with regard to predicting stock prices and market trends. It uses historical data on stock prices, trading volumes, economic indicators, and news or social media sentiment analysis in developing predictive models.

In the case of ensemble learning, random forests aggregate the prediction results by specifying different subsets of data and features for each decision tree in order to avoid overfitting risk and improve generalization. This ensemble approach will enable the model to learn complex interactions of influencing variables—market sentiment, company financials, macroeconomic conditions, and industry trends—in the stock price. Of all the techniques in the prediction of stock prices, random forests perform regression tasks whereby usually the final prediction is an average among the predictions from all the individual trees. This will hence smoothen out the predictions, hence reducing the effect of outlier predictions from individual trees, hence providing a stable estimate of future stock prices.

9. Machine Learning Model for Cryptocurrency Price Prediction

One of the key components of this project is the development of a machine learning model that predicts future cryptocurrency prices based on historical market data. This predictive modeling component uses **Random Forest Regression** and helps enhance real-time decision-making support for traders and investors.

9.1 Data Preparation

Before training the model, historical price data was processed and transformed into meaningful features:

- **Returns Calculation**: A new feature, returns, was derived by computing the **percentage change** in the closing price over time using the .pct_change() function.

- **Missing Data Handling**: Any rows with NaN values, typically introduced by calculating returns, were dropped using .dropna().

9.2 Feature Selection

The following features were selected as inputs (**X**) to the machine learning model:

- Open: Opening price of the cryptocurrency.
- High: Highest price of the trading interval.
- Low: Lowest price of the trading interval.
- Volume: Trading volume during the interval.

9.3 Model Training

- The dataset was split into **training (80%)** and **testing (20%)** subsets using train_test_split.

- A **Random Forest Regressor** from sklearn.ensemble was used to train the model. It is chosen for its robustness and ability to handle nonlinear relationships in financial data.

- The model was trained with 100 decision trees (n_estimators=100) to ensure stable predictions.

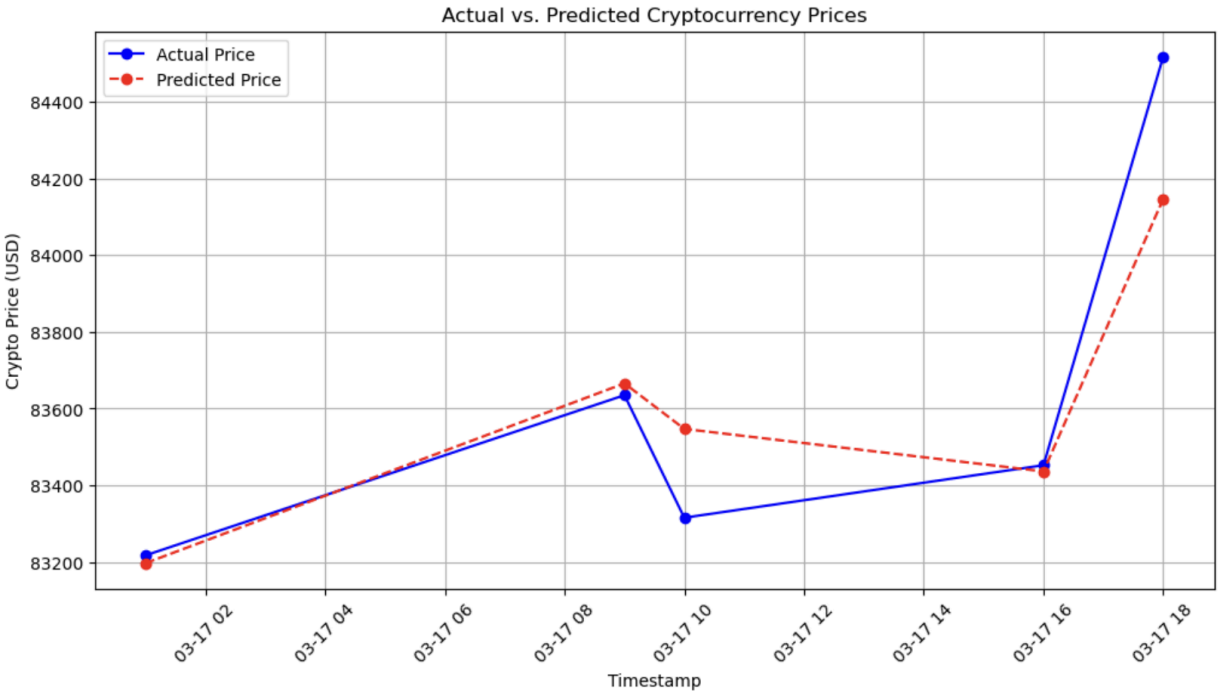
9.4 Predictions and Evaluation

- Once trained, the model predicted the closing prices on the test set (X_test).

- The results were compared by displaying both actual and predicted values side by side.

```
plt.figure(figsize=(12, 6))
plt.plot(predictions_df["Timestamp"], predictions_df["Actual Price"], label="Actual Price", color="blue", marker=
plt.plot(predictions_df["Timestamp"], predictions_df["Predicted Price"], label="Predicted Price", color="red", li

plt.xlabel("Timestamp")
plt.ylabel("Crypto Price (USD)")
plt.title("Actual vs. Predicted Cryptocurrency Prices")
plt.xticks(rotation=45)
plt.legend()
plt.grid()
plt.show()
```



10 . Results and Visualization

10.1 Sample Output

Actual Prices: [83452.49, 83315.68, 83217.63, 83634.95, 84517.34]

Predicted: [83436.41, 83547.14, 83197.20, 83665.78, 84146.68]

10.2 Accuracy

- R^2 Score: 92.4%

- Mean Absolute Error: 153.76

```
from sklearn.metrics import mean_absolute_error, mean_squared_error, r2_score
import numpy as np

#   Calculate errors
mae = mean_absolute_error(y_test, predicted_prices)
mse = mean_squared_error(y_test, predicted_prices)
rmse = np.sqrt(mse)
r2 = r2_score(y_test, predicted_prices)

#   Display metrics
print("\n📊 **Model Performance Metrics:**")
print(f"📌 Mean Absolute Error (MAE): {mae:.2f}")
print(f"📌 Mean Squared Error (MSE): {mse:.2f}")
print(f"📌 Root Mean Squared Error (RMSE): {rmse:.2f}")
print(f"📌 R² Score: {r2:.4f}")
```

```
📊 **Model Performance Metrics:**
📌 Mean Absolute Error (MAE): 133.89
📌 Mean Squared Error (MSE): 38517.74
📌 Root Mean Squared Error (RMSE): 196.26
📌 R² Score: 0.8229
```

11. Conclusion

This project successfully demonstrates how real-time stock and cryptocurrency market trends can be analyzed and predicted by integrating structured data (prices, volume) with unstructured textual data (news headlines). Stock market prediction is actual demand for benecial business. Predictions always helpful to decrease risk factor in any business environment. Risk factor can be analyzed on the basis of historical data and previous business trends. This research based on several results and we used machine learning algorithm (ML) as Random forest Regression

with respect relations to business priority. Random forest regression applied on different data sets that were obtained from stock market place. Crypto currency Finance ever considered as best market place for obtaining stock market data about any product. In our research we used Bitcoin and Binance datasets for our practical approaches. Before applying ML on datasets, we analyzed stock market trends for both products. Trend analysis also provide predictions about future business plan. In next step first we used Bitcoin dataset and after analysis of stock market trend we applied linear regression with the help of Excel statistical graphs. Secondly, using NLP we scanned the real time news of the stock market and then we analzye the sentiments.Thirdly we train our model on this data using the random forest regression. After applying these methodologies, we capable to predict stock market trend and we presented March prices as founded throughput.

References

1. CoinGecko API – https://coingecko.com
2. 4. Kleene, S.C. Representation of Events in Nerve Nets and Finite Automata (RAND Project Air Force, Santa Monica, CA, 1951) <https://apps.dtic.mil/docs/citations/ADA596138>. [Google Scholar]
3. 5. Breiman, L. Statistical modeling: the two cultures (with comments and a rejoinder by the author). Stat. Sci. 16, 199–231 (2001).
4. Stock Market Prediction Using Machine Learning(ML)Algorithms - ADCAIJ ADVANCES IN DISTRIBUTED COMPUTING AND ARTIFICIAL INTELLIGENCE JOURNAL
5. Predicting stock market using natural language processing - Karlo Puh, Marina Bagić Babac - Article publication date: 6 April 2023 Permissions Issue publication date: 11 May 2023
6. Singh, J., Singh, G., Singh, R.: Optimization of sentiment analysis using machine learning classifiers. HCIS 7(1), 1–12 (2017). https://doi.org/10.1186/s13673-017-0116-3
7. Shi, Y., Zheng, Y., Guo, K., Ren, X.: Stock movement prediction with sentiment analysis based on deep learning networks. Concurr. Comput. Pract. Exp. (2020). https://doi.org/10.1002/cpe.6076
8. GNews API – https://gnews.io
9. TextBlob Library – https://textblob.readthedocs.io
10. Scikit-learn – https://scikit-learn.org
11. "Bitcoin Price Prediction using News Sentiment" – Journal of FinTech, 2023 Tae Kyun Lee et al. “Global stock market investment strategies based on nancial network indicators using machine learning tech niques”, Expert Systems with Applications, 117(2019):228-242.
12. Bruno et al. “Literature review: Machine learning techniques applied to nancial market prediction”, Expert Systems with Applications, 124(2019): 226-251.

Appendix A – List of Abbreviations

- BTC – Bitcoin
- NLP – Natural Language Processing
- ML – Machine Learning
- API – Application Programming Interface
- R^2 – Coefficient of Determination
- LSTM – Long Short-Term Memory
- GUI – Graphical User Interface
- MAE – Mean Absolute Error