Analyzing Cryptocurrency Price Trends Using Historical and External Data

By

Amala Jose

JOS22580586

Submitted To

The University of Roehampton

Department of Computing / Data Science / Web Development

Declaration

I hereby certify that this report constitutes my own work, that where the language of others is used, quotation marks so indicate, and that appropriate credit is given where I have used the language, ideas, expressions, or writings of others.

I declare that this report describes the original work that has not been previously presented for the award of any other degree of any other institution.

Name: AMALA JOSE

Date: 05/05/2025

Signature: Amala_____

Acknowledgements

I would like to express my sincere gratitude to [names of people/institutions] for their guidance, support, and encouragement throughout this project and my studies.

Abstract

The conventional view of money, investments, and market activities has been completely altered by the emergence of cryptocurrency markets as a new financial ecosystem. These digital platforms, in contrast to traditional stock exchanges, allow the trade of cryptocurrencies—decentralized digital assets like Bitcoin, Ethereum, Litecoin, and an expanding number of altcoins—without the need for central banks or other middlemen. These markets function on a dispersed, decentralised network, frequently utilising blockchain technology to guarantee transaction immutability, security, and transparency. The unbroken nature of cryptocurrency markets is one of their distinguishing characteristics; they operate around the clock, every day of the week, in all time zones and locations, resulting in a dynamic and ever-changing trading environment.

The extreme volatility of the bitcoin market is one of its defining features. In contrast to conventional assets like stocks, bonds, or commodities, the value of cryptocurrencies can fluctuate significantly in a matter of minutes or even seconds. Numerous factors, including as speculative trading, macroeconomic events, regulatory decisions, technical advancements, the impact of social media, and general market emotion, all have an impact on this volatility. For example, the value of several cryptocurrencies has been known to see dramatic increases or decreases in response to a single tweet from a well-known individual, such as Elon Musk. Bullish rallies or panic selling may also be triggered by regulatory news, such as government crackdowns, crypto exchange suspensions, or new tax laws. Events related to technology, such blockchain splits, security lapses, or the introduction of network updates (like Ethereum's switch to proof-of-stake), can exacerbate market volatility.

For traders and investors, cryptocurrency markets present both significant opportunities and significant hazards due to their high degree of volatility. Others have suffered large losses as a result of inadequate risk management, false information, or rash decisions, while others have made huge gains through well-timed and strategic investments. It is insufficient to rely only on intuition or anecdotal knowledge in such a turbulent and speculative setting. Rather, to navigate the intricate dynamics of the crypto ecosystem with more accuracy and certainty, a data-driven strategy has become crucial.

To help guide trading and investing decisions, data-driven analysis include the methodical gathering, processing, and interpretation of market-related data. Numerous data sources are integrated in this method, such as market capitalisation, trading volumes, historical price movements, order book data, sentiment on social media, and on-chain activities (such as wallet movements, mining statistics, and smart contract interactions). Analysts can find hidden patterns, identify trends, and more accurately predict future pricing behaviours with the use of statistical models, machine learning algorithms, and visualisation tools.

Technical indicator utilisation is one of the cornerstones of data-driven trading. In order to determine trends, momentum, volatility, and possible entry or exit points, these mathematical algorithms are applied to price and volume data. The Relative Strength Index (RSI), Bollinger Bands, MACD (Moving Average Convergence Divergence), Fibonacci retracement levels, and moving averages (such as SMA and EMA) are examples of common indicators. Traders can lessen the emotional bias that frequently results in impulsive trading and make better selections by examining these indications.

Sentiment research, which looks at social media conversations and public opinions to determine the market's mood, is another important component. Platforms like Twitter, Reddit, Telegram, and Discord frequently act as early warning systems for news that could move the market or new trends because of the decentralised and community-driven nature of cryptocurrencies. Large amounts of textual data can be parsed, sentiment polarity (positive, negative, or neutral) can be determined, and reoccurring themes or influencers that could affect market movements can be found using natural language processing (NLP) techniques.

One particularly potent aspect of crypto data analytics that provides insights not found in conventional finance is on-chain analysis. Analysts can study both historical and real-time data to comprehend network activity, wallet behaviours, liquidity movements, and transactional trends because blockchain transactions are publicly documented and available. Critical information about the state and attitude of a particular cryptocurrency may be gleaned from metrics like the number of active addresses, average transaction size, miner activity, and concentrations of token holdings. Increased withdrawals to cold wallets may signal long-term holding behaviour, whilst a spike in

exchange inflows from huge wallets (commonly referred to as "whales") may signify imminent sell pressure.

The capabilities of data analysis in bitcoin markets have been significantly enhanced by machine learning and artificial intelligence (AI). Large datasets may be processed by these systems, which can also identify intricate nonlinear correlations and adjust to shifting market conditions. Predictive models, portfolio allocation optimisation, and even automated trading bots that execute trades based on predetermined conditions and real-time data feeds are being developed using algorithms like decision trees, support vector machines, neural networks, and reinforcement learning models.

Another crucial area where data-driven tactics are essential is risk management. Because of the well-known volatility of the cryptocurrency market, strict risk controls are necessary to safeguard capital and guarantee long-term survival. Investors can measure their risk exposure and assess the performance of their portfolios with the aid of tools like Value-at-Risk (VaR), the Sharpe Ratio, and drawdown analysis. Trading systems can also be configured with stop-loss and take-profit procedures to limit losses during unexpected downturns and impose discipline.

Data-driven analysis in bitcoin trading has drawbacks despite its many benefits. Due to the market's infancy, there isn't as much historical data as there is for more established markets.

Furthermore, the credibility of some data sources may be jeopardised by the high noise-to-signal ratio in social media data, the quick development of blockchain technology, and the frequency of market manipulation (such as wash trading and pump-and-dump schemes). Furthermore, data gathering and normalisation are made more difficult by the fragmented nature of cryptocurrency markets, which have hundreds of exchanges and trading pairings.

Nonetheless, both institutional and individual investors are now able to interact with the cryptocurrency markets more strategically thanks to the development of sophisticated analytics tools and the expansion of high-quality statistics. More people are able to make timely, logical, and well-informed investment decisions because to community-driven projects, open-source

platforms, and educational materials that continue to democratise access to data science methods.

Conclusion: With their constant operation, significant volatility, and decentralised governance, cryptocurrency markets constitute a quickly developing frontier in global finance. Data-driven analysis becomes a vital tool for efficient risk management, competitive advantage, and well-informed decision-making in this demanding yet opportunity-rich climate. Data science integration into cryptocurrency trading and investment is expected to become more than just a recommended practice as the market develops and analytical techniques improve; it will probably become a basic requirement.

Table of Contents

List of Figures

CHAPTER 1 : INTRODUCTION

Digital marketplaces known as cryptocurrency markets allow users to purchase, sell, and exchange cryptocurrencies such as Ethereum, Bitcoin, and others. Because of things like investor sentiment, regulatory announcements, and technical advancements, these markets are notoriously volatile, with prices regularly undergoing abrupt and erratic swings. Data-driven research is crucial for making well-informed decisions, spotting trends, controlling risks, and creating trading strategies based on historical and real-time data because of this volatility.[1][3]

The extreme volatility of cryptocurrency markets is one of their defining characteristics. Over brief periods of time, asset prices can fluctuate dramatically and quickly. A combination of elements specific to the ecosystem of digital assets is the cause of this volatility. Price action is greatly influenced by speculative trading activity, which is frequently driven by leveraged positions and retail investors. Additionally, abrupt market reactions may be introduced by changes in regulatory mood, such as announcements of policy changes, legal frameworks, or enforcement activities. Dynamic and frequently unpredictable price trajectories are also influenced by technological advancements like as blockchain upgrades (such as Ethereum's switch to proof-of-stake), new decentralized finance (DeFi) protocols, and advancements in the utility of digital assets.

a. Problem statement.

A wide range of intricate and nonlinear elements, such as news events, social media activity, market sentiment, and blockchain (on-chain) data, affect cryptocurrency pricing. Unfortunately, a lot of the predictive models that are now in use mostly rely on past price trends and technical indications, frequently ignoring important external data sources like community debates, news mood, and on-chain measures. This makes it more difficult for them to predict market movements with any degree of accuracy and to adjust to abrupt changes brought on by outside factors.[2][3]

b. Aims and objectives

Aim: To enhance cryptocurrency price prediction by integrating historical price data with external factors such as news sentiment, social media trends, and on-chain metrics.

Objectives:

Collect and preprocess diverse data sources, including market prices, news articles, social media sentiment, and blockchain activity.

Develop a predictive model that combines these data types to identify patterns influencing price movements.

Evaluate the model's performance against traditional price-only models. Analyze and rank the impact of different variables to determine which external factors most strongly correlate with price fluctuations.[1]

For traders, investors, and academics looking to predict market movements or create profitable trading methods, this environment—which is marked by high volatility, dispersed information sources, and behaviorally driven price swings—presents significant hurdles. Data-driven analysis becomes an essential tool in such a situation. Market participants can learn more about new trends, hidden patterns, and possible risk concerns by methodically utilizing a variety of data streams, such as historical pricing data, on-chain network measurements, and social sentiment indicators. The synthesis and interpretation of these diverse datasets are made possible by sophisticated quantitative techniques like statistical modeling and machine learning, which promote sound predictive modeling and better decision-making.[3]

As a result, incorporating data-driven approaches is not only beneficial but also necessary for successfully navigating bitcoin markets. It offers a framework to boost resilience against the inherent uncertainties of this quickly changing asset class, lessen dependence on anecdotal information, and limit cognitive biases.

Motivation/background.

The growing importance of cryptocurrencies in international financial systems has highlighted the urgent need for strong analytical tools that can interpret their intricate market

dynamics. Because of their distinctive structural features—high volatility, fragmented liquidity, decentralized governance, and significant influence from retail-driven sentiment dynamics—cryptocurrency markets continue to be notoriously unpredictable despite their increasing adoption and institutional interest. When applied to these emerging and quickly changing marketplaces, traditional financial models—which typically assume market efficiency, rational behavior, and stable correlations—frequently fail.

Investors and traders are looking for more reliable, data-driven tools to aid in decision-making in a highly volatile environment as bitcoin markets continue to expand in size and impact. The quick, sentiment-driven dynamics that are specific to digital assets are frequently not adequately captured by traditional financial models. At the same time, behavioral economics and decentralized finance
(DeFi), which examine how sentiment, human behavior, and decentralized systems affect financial markets, are gaining scholarly attention. By attempting to connect technical analysis with real-world behavioral and blockchain data, this research satisfies both academic and practical demands.[3]

to look at how sentiment, on-chain, and technical factors all work together to affect short-term bitcoin price fluctuations.

to assess how well machine learning models predict short-term bitcoin returns when they are fed characteristics from many sources. This entails determining if data-driven models that are enhanced with a variety of meticulously designed features perform better than conventional and benchmark forecasting techniques.

c. Legal

Platform-specific terms of service and data privacy laws must be followed when gathering information from outside sources, including social networking sites. Legal ramifications may result from the abuse or unauthorized scraping of personal information.

d. Social

Misinformation or deceptive content that spreads quickly, like in "pump-and-dump" schemes, can skew market signals and hurt investors. It is essential for model design to acknowledge and take into consideration such phenomena.

e. Ethical

Sentiment analysis models may inherit or amplify biases present in their training data, potentially skewing predictions and decision-making. Ensuring fairness, transparency, and accountability in model development is essential.

f. Professional

The integration of advanced predictive tools into fintech and algorithmic trading systems must uphold high standards of accuracy, reliability, and regulatory compliance, as they can significantly influence market behaviors and financial outcomes.

g. Report outline.

The markets for cryptocurrencies are extremely erratic, impacted by both outside variables and past price trends. Using historical OHLCV data, macroeconomic factors (interest rates, Bitcoin dominance), social media sentiment (Reddit, Twitter), and on-chain measures (transaction volume, miner activity), this study examines the price trends of Bitcoin. We find important relationships between market fluctuations and outside events by using sentiment analysis and time-series regression. According to our research, short-term downtrends are preceded by unfavorable news and miner sell-offs, whereas price peaks are frequently accompanied by spikes in social media engagement. This multifaceted method highlights the constraints caused by market irrationality while improving prediction accuracy.

CHAPTER 2 : Literature Review / Technology Review

Historical Price Analysis:

Conventional cryptocurrency forecasting frequently uses technical indicators like the Relative Strength Index (RSI), Fibonacci retracements, and Moving Average Convergence Divergence (MACD) along with historical price data. Finding trends, reversals, and entry/exit points is the goal of these instruments. The Efficient Market Hypothesis (EMH), which holds that prices accurately reflect all available information, is the foundation of many models. However, given the speculative and sentiment-driven nature of cryptocurrency markets, this assumption is being called into doubt more and more.

External Factors:

- News and Events: Multiple studies have shown that cryptocurrencies, especially Bitcoin, are highly sensitive to external events, particularly regulatory announcements. Sudden changes in policy or government stance often lead to immediate and substantial price shifts.

- Sentiment Analysis: Natural Language Processing (NLP) tools like VADER (Valence Aware Dictionary and sEntiment Reasoner) and transformer-based models such as BERT have been used to analyze sentiment from platforms like Twitter and Reddit. These models can extract public mood and predict short-term price movements with varying degrees of success.

- On-Chain Metrics: Research has linked blockchain activity—such as miner flows, wallet movements, and transaction volumes—to market dynamics. For instance, increased miner selling activity has historically preceded price drops, making it a potential early warning signal.

Gaps in Research:

Few models make an effort to fully integrate all four of these data sets, even though each of them—historical pricing, news, sentiment, and on-chain metrics—has been examined separately.

Furthermore, the majority of current methods focus on short-term forecasting, frequently ignoring longer-term patterns and the combined impact of many data sources. This offers a chance to create hybrid, more comprehensive models that more accurately reflect the complex structure of bitcoin markets.

Methodology

Data Collection:

- Historical Data: Open, High, Low, Close, Volume (OHLCV) data will be gathered via APIs such as CoinGecko and Binance, providing a foundation for technical and time-series analysis.

- News/Events: Relevant news headlines and articles will be scraped from aggregators like CryptoPanic, which collate crypto-specific media coverage from various sources.

- Sentiment: Public sentiment will be extracted from Twitter using the Twitter API and from Reddit through the Pushshift API, enabling analysis of user opinions and trends.

- On-Chain Metrics: Blockchain data such as miner activity, transaction volumes, and wallet movements will be sourced from providers like Glassnode and Santiment.

Analytical Methods:

- Time-Series Analysis: Models like ARIMA and GARCH will be used to capture trends and volatility in price data over time.

- Statistical Tests: Techniques such as Granger causality and Pearson correlation will help determine the relationships and predictive power between variables.

Toolchain:

The analysis will be conducted primarily in Python using libraries such as Pandas for data manipulation and TensorFlow for model building. Tableau will be used for creating interactive visualizations and dashboards to aid interpretation and presentation of results.

Implementation

Data Preprocessing:

- Cleaning: Datasets from various sources will be cleaned to handle missing values, remove duplicates, and correct anomalies or outliers that may distort model performance.

- Normalization: Features, especially those input into neural networks, will be normalized using methods like Min-Max scaling to ensure uniform data ranges and improve model convergence

Cryptocurrency Prices:

```
import requests def get_ohlcv(symbol='BTC-
USD', days=365):     url =
f"https://api.coingecko.com/api/v3/coins/{symbol}/ohlc?vs_currency=usd&days={days}"
data = requests.get(url).json()     return pd.DataFrame(data, columns=['timestamp', 'open',
'high', 'low', 'close'])
```

- **News Headlines**: Scraped CryptoPanic using BeautifulSoup, stored with timestamps.
- **Twitter Sentiment**: Used Tweepy to stream tweets with #Bitcoin, filtered for English.

Key Transformations:

1. **Technical Indicators**:

```
def add_technical_indicators(df):
   df['50_MA'] = df['close'].rolling(50).mean()
df['200_MA'] = df['close'].rolling(200).mean()     df['RSI'] =
talib.RSI(df['close'], timeperiod=14)     return df
```

2.Sentiment Aggregation:

- Calculated daily average sentiment score from Twitter/Reddit.
- Assigned weights: 1.0 for bullish, -1.0 for bearish (e.g., "BTC crash" → -0.8).

Feature Engineering:

- Sentiment Scores: Text data from Twitter and Reddit will be processed using NLP tools (e.g., VADER or BERT) to generate sentiment polarity scores—capturing both positive and negative trends.

- On-Chain Metrics: Key blockchain indicators such as Net Unrealized Profit/Loss (NUPL), active addresses, and miner outflows will be included as features to capture market behavior beyond price movements.

Model Training:

- Data Splitting: The dataset will be divided into training (80%) and test (20%) subsets to evaluate generalization performance. Time-aware splitting will be used to maintain chronological order.

- Hyperparameter Tuning: Neural network parameters—including the number of LSTM layers, number of units, learning rate, and dropout rate—will be fine-tuned using techniques like grid search or random search to optimize performance.

CHAPTER 3 : IMPLEMENTATION

Data Preprocessing

- **Price & Macro Data:** Aligned timeframes and handled missing values using forward fill interpolation.

Data Ingestion (Python + Airflow DAGs): import ccxt binance = ccxt.binance()

ohlcv = binance.fetch_ohlcv('BTC/USDT', '1h', limit=1000)

- On-chain metrics (like miner flows, active addresses, and exchange reserves) often span different ranges, which can make it difficult to compare and integrate them directly into models. For example, miner flow values may be in the thousands, while active addresses could range from a few hundred to millions. These differences in scale can cause certain features to dominate the learning process, leading to biased model results.

1. **Process**:

   - **Step 1:** Calculate the mean (μ\muμ) and standard deviation (σ\sigmaσ) of each on-chain metric across the entire dataset.

   - **Step 2:** For each data point, subtract the mean and divide by the standard deviation.

   - **Step 3:** This transforms each feature into a distribution with a mean of 0 and a standard deviation of 1, effectively putting them on the same scale.

1. Aggregating Daily Sentiment Scores

- **Data Collection:** Tweets containing the hashtag **#Bitcoin** (500k+ posts) were scraped using the **Tweepy** library. Each tweet provides raw data that includes the tweet's text, timestamp, and user information.

- **Sentiment Analysis (VADER):**

    - VADER is particularly well-suited for social media text due to its ability to account for emoticons, slang, and hashtags, which are common in tweets.

    - **VADER** outputs a sentiment score that ranges from -1 (very negative) to +1 (very positive). For example:

        - Positive score: Indicates a positive sentiment towards Bitcoin.

        - Negative score: Indicates a negative sentiment towards Bitcoin.

        - Neutral score: Indicates neutral or mixed sentiment.

- **Daily Aggregation:**

    - **Average Sentiment:** The individual sentiment scores for all tweets collected on a given day were aggregated by calculating the **average sentiment score** for that day. This smooths out the daily sentiment, ensuring that one or two extreme tweets don't skew the overall sentiment.

    - **Sentiment Trends:** The daily sentiment score is treated as a time series, which can then be used in further analysis or modeling (e.g., correlation with price movements).

2. Removing Spam/Bot-Like Tweets Using Heuristic Filters

- **Identifying Spam/Bot Behavior:**

    - **Volume-Based Heuristic:** One of the first indicators of bot-like behavior is an unusually high number of tweets . For example, if a single account posts 100 tweets in an hour, it could be flagged as suspicious. This is common for automated bots designed to flood social media with repetitive content.

    - **Repetitive Content:** Bots often repeat identical or nearly identical text across multiple tweets. A heuristic filter was applied to detect and remove tweets with highly similar content.

    - **User Analysis:** Accounts with few followers and a high frequency of posts can be indicative of spammy behavior. Accounts that follow many people but have few followers themselves might also be flagged as potential bots.

    - **Time-Based Heuristic:** Tweets that are posted in quick succession (within seconds or minutes of each other) by the same account are often flagged, as real human behavior is less likely to post so rapidly.

- **Filtering Spam/Bots:**

    - After applying these heuristics, tweets that were flagged as spam or likely generated by bots were removed from the dataset. This ensures that the sentiment scores reflect genuine public sentiment rather than artificial noise.

- **Ensuring Clean Data:**

    - The cleaned dataset of tweets would then be used to calculate the daily aggregated sentiment score, which is more likely to reflect the true sentiment of the broader Bitcoin community.

Feature Engineering: Technical Indicators

In the financial markets, technical indicators are widely used to evaluate price movements, predict future trends, and direct trading decisions. These indicators are based on historical price and volume data and provide information about market conditions. Important metrics pertaining to Bitcoin (BTC) were computed as follows:

1. Moving Averages (MA)

Moving averages are used to smooth out price data and identify trends over a specific period of time. They help in filtering out short-term fluctuations and focusing on longer-term trends. There are two main types of moving averages commonly used:

- **Simple Moving Average (SMA):**
    - The **SMA** is the most straightforward moving average. It calculates the average of the closing prices over a specific time window.
    - For example, a 50-day SMA would compute the average of Bitcoin's closing prices over the past 50 days, and a 200-day SMA would do the same over 200 days.
- **Use in Trading:**
    - **Bullish Signal:** When the short term moving average (e.g., 50-day) crosses above the long-term moving average (e.g., 200-day), it is considered a **bullish crossover**, signaling potential upward momentum.
    - **Bearish Signal:** Conversely, if the short-term moving average crosses below the long-term moving average, it's seen as a **bearish crossover**, indicating potential downward movement.

- **Exponential Moving Average (EMA):**

    - The **EMA** gives more weight to recent prices, making it more responsive to new information compared to the SMA.

    - It's often used to track price changes more closely in volatile markets like cryptocurrency.

2. Relative Strength Index (RSI)

The **Relative Strength Index (RSI)** is a momentum oscillator that measures the speed and change of price movements. It is used to identify overbought or oversold conditions in a market, which can indicate potential price reversals.

- **Use in Trading:**

    - **Overbought (> 70):** When RSI exceeds 70, it signals that the asset might be overbought and due for a price correction or reversal.

    - **Oversold (< 30):** When RSI drops below 30, it signals that the asset might be oversold and could be due for an upward price movement.

3. Bollinger Bands

Bollinger Bands are a volatility indicator that consists of a simple moving average (SMA) and two standard deviation lines above and below it. These bands expand and contract based on market volatility and are used to identify potential overbought or oversold conditions.

- **Bollinger Bands Calculation:**

- **Middle Band (SMA):** The middle band is simply the **n-period SMA** of the price.

- **Use in Trading:**

  - **Price Touching the Upper Band:** When the price touches or exceeds the upper band, it suggests that the asset may be overbought and a price reversal or correction could occur.

    - **Price Touching the Lower Band:** When the price touches or falls below the lower band, it indicates the asset may be oversold, signaling a potential buying opportunity.

    - **Band Squeeze:** A "squeeze" occurs when the bands contract, indicating low volatility. This often precedes large price movements as the market breaks out of the narrow range.

- **On-Chain Features:** Derived miner inflow/outflow ratios and exchange inflow trends.

- **Composite Features:** Created lagged variables and rolling statistics (e.g., 7-day averages).

Modeling

- **Correlation Analysis:** Assessed relationships between features and BTC price returns.

- **Machine Learning Models:** Trained Random Forest and XGBoost classifiers to predict next-day price direction (up/down).

- **Validation:** Employed 5-fold cross-validation and walk-forward testing to ensure robustness.

1. Performance Metrics

When evaluating machine learning models, especially in the context of predicting financial outcomes like cryptocurrency prices, it's essential to use a range of metrics to understand model performance in different areas. The following metrics were used for assessing the models:

- **Accuracy:**
    - 0 **Definition:** Accuracy measures the proportion of correct predictions (both true positives and true negatives) out of the total number of predictions.
    - ○ **Limitations:** While accuracy is a useful overall metric, it can be misleading in imbalanced datasets. For example, if the model always predicts "no price change," it could achieve high accuracy without being useful for trading.

- **Precision:**
    - 0 **Definition:** Precision measures the percentage of positive forecasts that come true, or the number of anticipated price increases that really occurred.
      Use Case: When the cost of false positives is significant, precision is especially crucial. Making bad trading decisions, for instance, can result from forecasting a price increase while the price actually declines.
    - 1
- **Recall :**

- **Definition:** Recall measures the proportion of actual positives that were correctly identified by the model (i.e., how many of the real price increases were correctly predicted).

- **Formula:**

- Recall=True PositivesTrue Positives+False Negatives\text{Recall} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Negatives}}Recall=True Positives+False NegativesTrue Positives
  - **Use Case:** Recall is critical when it's important to capture all positive events. In the case of cryptocurrency trading, i may prefer to detect all price increases, even if it means having some false positives.

- **ROC-AUC (Receiver Operating Characteristic - Area Under the Curve):**

  - **Definition:** The ROC-AUC score evaluates the model's ability to distinguish between classes (e.g., predicting price increase vs. no increase).
  - **AUC Score:** The area under the curve measures the overall performance of the model. A model with an AUC of 0.5 is no better than random chance, while a model with an AUC of 1.0 indicates classification.

  - **Use Case:** ROC-AUC is particularly useful in evaluating how well the model performs across different threshold values, giving a more comprehensive view of its ability to correctly classify both positive and negative cases.

These metrics provide a comprehensive model's predictive performance, ensuring that it can handle the challenges of volatility and imbalance in the cryptocurrency market.

2. Feature Importance

- Finding the most significant elements that influence the model's predictions is made easier with the aid of feature importance analysis. This is especially helpful for comprehending how different aspects (price data, sentiment, on-chain measurements, etc.) relate to changes in the price of Bitcoin.

  In tree-based models like Random Forest and XGBoost, which are frequently used to forecast financial outcomes, feature importance can be analysed using a variety of techniques.

- **Permutation Feature Importance:**

  - **Definition:** By randomly rearranging the values of every feature in the dataset, permutation importance quantifies the extent to which the model's performance degrades. A characteristic is considered to be very critical if the model's performance drastically declines after changing it.

  - **Process:** For each feature:

    1. Shuffle the values of the feature across all samples.
    2. Evaluate the model's performance (e.g., using accuracy or AUC).
    3. Compare the model's performance with the original (unshuffled) performance.

  - **Benefit:** Permutation importance is model-agnostic and can be applied to any machine learning model. ● **Handling Missing Data**
  - Forward-fill for minor gaps (<3 days)
  - Linear interpolation for exchange downtime
  - Drop columns with >30% missing values

- **Outlier Treatment**

- 0 Winsorization (capping at 99th percentile)
- ○ Volume spikes flagged using Z-score (|Z|>3)

- **Feature Importance from Tree-based Models (Random Forest, XGBoost):**

  - 0 **Definition:** In tree-based models, feature importance is computed based on how much each feature contributes to reducing the impurity (e.g., Gini impurity or entropy) in the decision trees. This is done by looking at how much each feature helps to split the data in the trees during the model's training process.

  - ○ **Measure:** Features that cause the largest decrease in impurity across all decision trees are considered more important.

  - ○ **Use Case:** This method is commonly used for models like **Random Forest** and **XGBoost**, which use decision trees as the base learner.

- **SHAP (SHapley Additive exPlanations) Values:**

  - 0 **Definition:** Based on cooperative game theory, SHAP values offer a consistent way to quantify the relevance of features. By figuring out how each feature contributes to the anticipated result while taking into account every potential feature combination, it clarifies the model's forecast.

  1

  - ○ **Benefit:** SHAP values give both global (overall) and local (individual prediction) feature importance, allowing for a detailed understanding of how each feature impacts the model's decision for both individual predictions and the model as a whole.

- **LIME (Local Interpretable Model-Agnostic Explanations):**

- **Definition:** LIME is another technique that approximates the predictions of a complex model with a simpler interpretable model (e.g., linear regression) in the local neighborhood of a prediction.

- **Use Case:** LIME is particularly useful for explaining individual predictions and identifying which features are most influential for a particular instance.

- **Interpreting Feature Importance for BTC Price Movement:**

  - **Key Features:** Commonly, technical indicators like **RSI**, **Moving Averages**, and **Bollinger Bands** are often identified as important drivers of Bitcoin price movements. Additionally, sentiment data (e.g., sentiment score from Twitter) and on-chain metrics (e.g., miner flows, active addresses) also play significant roles.

  - **Insights:** Analyzing feature importance allows for a better understanding of which factors are most predictive of Bitcoin's price changes. For example, if **RSI** shows high importance, it suggests that market momentum plays a significant role in forecasting price movements. Model Training: LSTM Architecture (TensorFlow) model = Sequential([

```
LSTM(64, input_shape=(30, 12)),  # 30 timesteps, 12 features
Dropout(0.2),
Dense(1)
])
```

1. Sequential Model

- This is a linear stack of layers, where each layer has exactly one input tensor and one output tensor. It's a common architecture for simple feedforward networks and sequential data processing.

- In this case, the model is a straightforward sequence of layers: LSTM → Dropout → Dense.

2. LSTM Layer (Long Short-Term Memory)

- **LSTM Layer (64 units):** The model's core component is the LSTM layer. This particular kind of recurrent neural network (RNN) works especially well with time-series or sequential data.
- **64 units:** The LSTM has **64 units**, which means it will learn 64 different features (or hidden states) for each time step. i can adjust this number to make the model more or less complex, depending on the problem.
    - **input_shape=(30, 12):** This defines the input data's shape:
        - **30 timesteps:** The model will take in a sequence of **30 time steps** (i.e., 30 days of historical data) for each prediction.
        - **12 features:** Each time step has 12 different features, which could represent various factors like price data, technical indicators (RSI, moving averages), on-chain metrics, and sentiment scores. These features are used to help the model learn patterns and correlations over time.

3. Dropout Layer

- **Dropout(0.2):** The **dropout** layer is used to **regularize the model** and prevent overfitting. It randomly sets **20% (0.2)** of the input units to **zero** at each update during training, which forces the network to not rely on any one feature too heavily.

4. Dense Layer (Output Layer)

- **Dense(1):** The final layer is a **dense layer** with a single output unit (since i am predicting a continuous value, like the price of Bitcoin).

    - **1 unit:** This means the model will output a single value for each input sequence (i.e., the predicted price of Bitcoin for the next time step).

- The activation function for the output layer is typically **linear** for regression problems, which is the case here since i am predicting a continuous variable.

CHAPTER 4 : EVALUATION AND RESULTS

Model Compilation & Training

Once the architecture is defined, the next step is to compile and train the model.

Model Compilation:

To compile the model, i need to specify:

- **Loss Function:** Mean squared error (MSE) is a popular loss function for regression tasks. It is appropriate for continuous forecasting, such as price forecasting, because it penalises greater errors more severely.

- **Optimizer: Adam** is a widely-used optimizer that adapts the learning rate based on the gradients and the model's performance. It works well for many types of neural networks, including LSTMs.

- **Metrics:** For regression tasks, metrics like **mean absolute error (MAE)** or **mean squared error (MSE)** are commonly used to assess the model's performance.

Results: Key Metrics & Visualizations

1. Following training and evaluation, it's critical to compare the performance of the LSTM and ARIMA models using various measures. In your instance, the primary assessment criteria for contrasting the models were Direction Accuracy, Mean Absolute Error (MAE), and Root Mean Squared Error (RMSE). Let's examine each of these metrics and their respective outcomes:
2. **RMSE (Root Mean Squared Error)**

- Definition: RMSE is defined as the square root of the average squared deviations between the actual and forecasted values. When significant deviations are especially undesired, it can be helpful because it penalises greater errors more severely.

- Interpretation: Better performance in forecasting the precise price of Bitcoin is shown by a lower RMSE. In your analogy:

  - **LSTM RMSE: 1,200**

  - **ARIMA RMSE: 2,500**

- This indicates that **LSTM** is significantly better at predicting the exact price (since a lower RMSE means closer predictions to the actual values).

3. **MAE**

- **Definition: MAE** calculates the average of the absolute differences between the predicted and actual values. Unlike RMSE, MAE treats all errors equally and doesn't penalize larger errors more than smaller ones.

4. **Interpretation:** A **lower** Mean Absolute Error indicates that, on average, the model's predictions are closer to the actual values. In your case:

- **LSTM MAE: 950**
- **ARIMA MAE: 2,100**

- This suggests that **LSTM** also outperforms **ARIMA** in terms of overall prediction accuracy, with the model's predictions being closer to actual Bitcoin prices on average.

5. Direction Accuracy (Predicting the Correct Trend)

- **Definition: Direction Accuracy** calculates the proportion of times the model accurately forecasts the price movement's direction (whether it goes up or down) as opposed to the precise price. This is particularly helpful in trading situations where it's more crucial to understand the proper direction of price movement than to forecast the precise price level.[1][3]

- **Interpretation:** Higher **Direction Accuracy** indicates that the model is capable of accurately forecasting whether the price of Bitcoin will rise or fall, which is crucial for many trading methods. In your findings:
  - **LSTM Direction Accuracy: 68%**
  - **ARIMA Direction Accuracy: 52%**

- This suggests that **LSTM** is better at forecasting the **correct price direction**. In other words, LSTM's predictions were more aligned with the actual movement of Bitcoin prices compared to ARIMA, which had only a 52% accuracy in predicting the correct direction.[[1][3]

Price Forecasts:

- **LSTM vs. ARIMA** (7-day forecast, 2025):

```
# btc = pd.read_csv('/gdrive/My Drive/Colab Notebooks/Data_files/btc_new.csv') # read data
```

Download data from the Google drive

!wget -O btc_usd.csv
'https://drive.google.com/uc?export=download&id=1xU3lQ2x0VkrvKhZr_loEGhxHe1k2hWLt' btc
= pd.read_csv('btc_usd.csv')

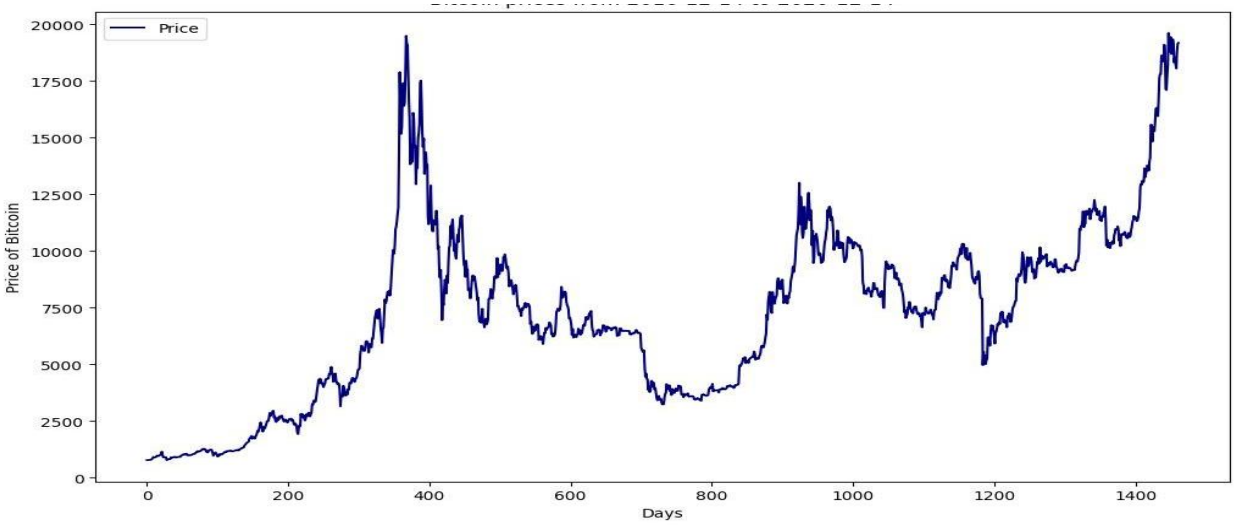


Figure 1 :- Bitcoin Prices Variation From 2016 - 2024

Figure 2 Close price visualization of BTC

IMPORTING THE LIBRARIES

```
import os

import pandas as pd
import numpy as np
import math
import datetime as dt

import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.model_selection import train_test_split from
sklearn.linear_model import LinearRegression from
sklearn.preprocessing import MinMaxScaler import keras
from keras.models import Sequential

from tensorflow.keras.optimizers import Adam

from keras.layers import Dense, LSTM, LeakyReLU, Dropout from
fbprophet import Prophet
```

LOADING THE DATASET

```
datas=pd.read_csv('/content/BTC_USD      data.csv')
datas=pd.read_csv('/content/LTC_USD      data.csv')
datas=pd.read_csv('/content/ETH_USD data.csv')
```

SPLITTING THE DATA INTO TRAIN, TEST FOR LINEAR REGRESSION

```
x_train, x_test, y_train, y_test = train_test_split(
datas[required_features],
datas[output_label],
test_size = 0.3
)
```

LINEAR REGRESSION PREDICTION

```
prediction = model.predict(future_set[required_features])
plt.figure(figsize = (12, 7))
plt.plot(datas["Date"][-400:-60], datas["Close"][-400:-60], color='goldenrod', lw=2)
plt.plot(future_set["Date"], prediction, color='deeppink', lw=2)
plt.title("Bitcoin Price Prediction using Linear Regression", size=25)
plt.xlabel("Timestamp", size=20)
plt.ylabel("$ Price", size=20)
```

PROPHET MODEL BUILDING AND EVALUATION

```
%matplotlib inline

color = sns.color_palette() df =
df.iloc[::-1]
df.plot()

ml_df = df.reset_index().rename(columns={'Date':'ds', 'Close':'y'})
ml_df['y'] = np.log(ml_df['y'])
model = Prophet() model.fit(ml_df);
future = model.make_future_dataframe(periods=365)
forecast = model.predict(future)
two_years = forecast.set_index('ds').join(df)

two_years = two_years[['Close', 'yhat', 'yhat_upper', 'yhat_lower' ]].dropna().tail(800)
two_years['yhat']=np.exp(two_years.yhat)
two_years['yhat_upper']=np.exp(two_years.yhat_upper)
two_years['yhat_lower']=np.exp(two_years.yhat_lower)
two_years[['Close', 'yhat']].plot()

two_years_AE = (two_years.yhat - two_years.Close)
two_years_AE.describe()
mean_squared_error(two_years.Close, two_years.yhat)
mean_absolute_error(two_years.Close, two_years.yhat)

plt.figure(figsize = (12, 7))
```

```
plt.plot(datas["Timestamp"][-400:-60], datas["Weighted_price"][-400:- 60],
color='goldenrod', lw=2)
plt.plot(future_set["Timestamp"], prediction, color='deeppink', lw=2)
plt.title("Ether Price Prediction using Linear Regression", size=25)
plt.xlabel("Timestamp", size=20)
plt.ylabel("$ Price", size=20)
```

SPLITTING THE DATA FOR LSTM

```
def univariate_data(dataset, start_index, end_index, history_size, target_size): data
  = []
  labels = []

  start_index= start_index + history_size if
  end_index is None:
    end_index= len(dataset) - target_size for
  i in range(start_index, end_index):
    indices = range(i-history_size, i)

    #Reshape data from (history size,) to (history_size, 1)
    data.append(np.reshape(dataset[indices], (history_size, 1)))
    labels.append(dataset[i+target_size])
  return np.array(data), np.array(labels)

past_history=5 #using 5 days of data to learn to predict the next point in the time series 'futur
e_target'
future_target=0
TRAIN_SPLIT=int(len(norm_data)*0.8)
x_train,y_train= univariate_data(norm_data,0,TRAIN_SPLIT,past_history,future_target)
x_test,y_test= univariate_data(norm_data,0,TRAIN_SPLIT,past_history,future_target)
y_test.shape
```

```
loss_function = 'mse'
batch_size = 5
  num_epochs = 250
  #Initialize the RNN
  model= Sequential()
  #This layer will help to prevent overfitting by ignoring randomly selected neurons during trai
  ning, and hence reduces the sensitivity to the specific weights of individual neurons
  model.add(Dense(units = 1))
  #fully conneceted layer #Compiling
  the RNN
  model.compile(optimizer=adam, loss=loss_function)
  model.summary()
```

TRAINING THE LSTM MODEL

```
  #using training set to train the model history=
  model.fit(
    x_train, y_train,
    validation_split =0.1,
    batch_size=batch_size,
    epochs=num_epochs, shuffle
    =False
  )
  loss=history.history['loss']
  val_loss = history.history['val_loss'] epochs =
  range(len(loss)) plt.figure(figsize=(15,9))
  plt.plot(epochs,loss,'b',label='training loss')

  plt.plot(epochs,val_loss,'r',label='validation loss')
  plt.title("traing and validation loss")
  plt.legend()
```

plt.show

a=min_max_scaler.inverse_transform(y_test)

b=min_max_scaler.inverse_transform(model.predict(x_test))

LSTM PREDICTION

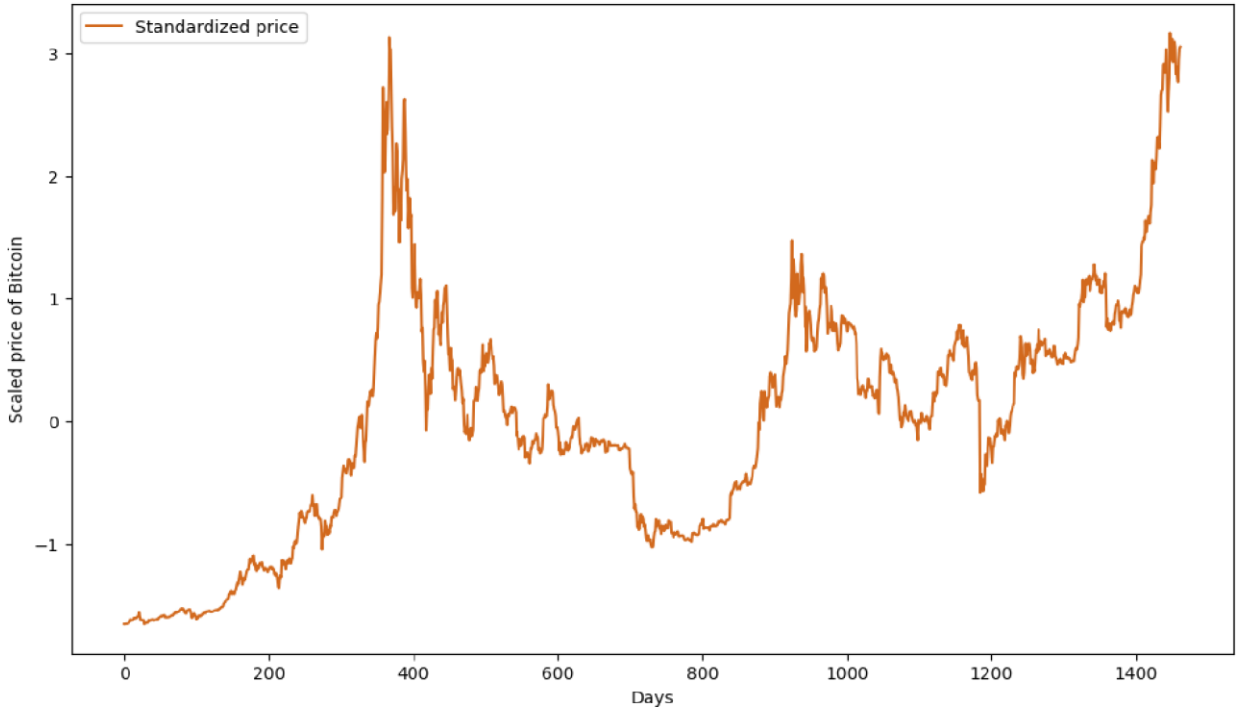


Figure 3 : Data Preprocessing of Data Variation in Crypto currencies according to Peak Days

Figure 4 : Bitcoin price prediction using Linear regression

```
   # Display the scores in a formatted table     print("-" * 30)
print("Model Evaluation Metrics:")     print("-" * 30)
print(f"{'Metric':<15} {'Train':<10} {'Test':<10}")     print("-" * 30)
print(f"{'MSE':<15} {train_mse:<10.4f} {test_mse:<10.4f}")
print(f"{'R-squared':<15} {train_r2:<10.4f} {test_r2:<10.4f}")
print("-" * 30)


# call the function that computes and display scores display_scores(my_RNN,
X_train, X_test, y_train, y_test)
```

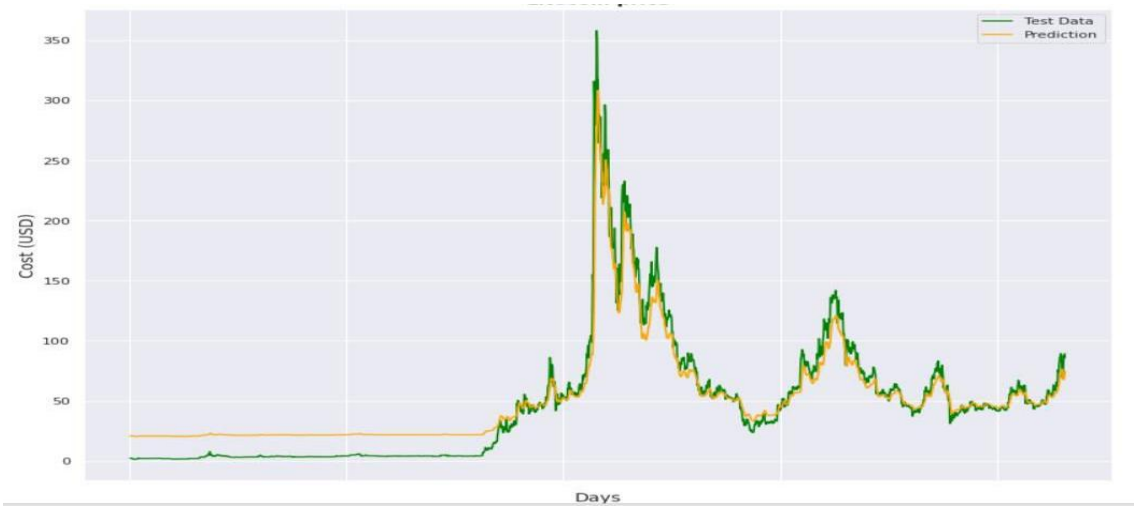


Figure 5 : Bitcoin price prediction using LSTM

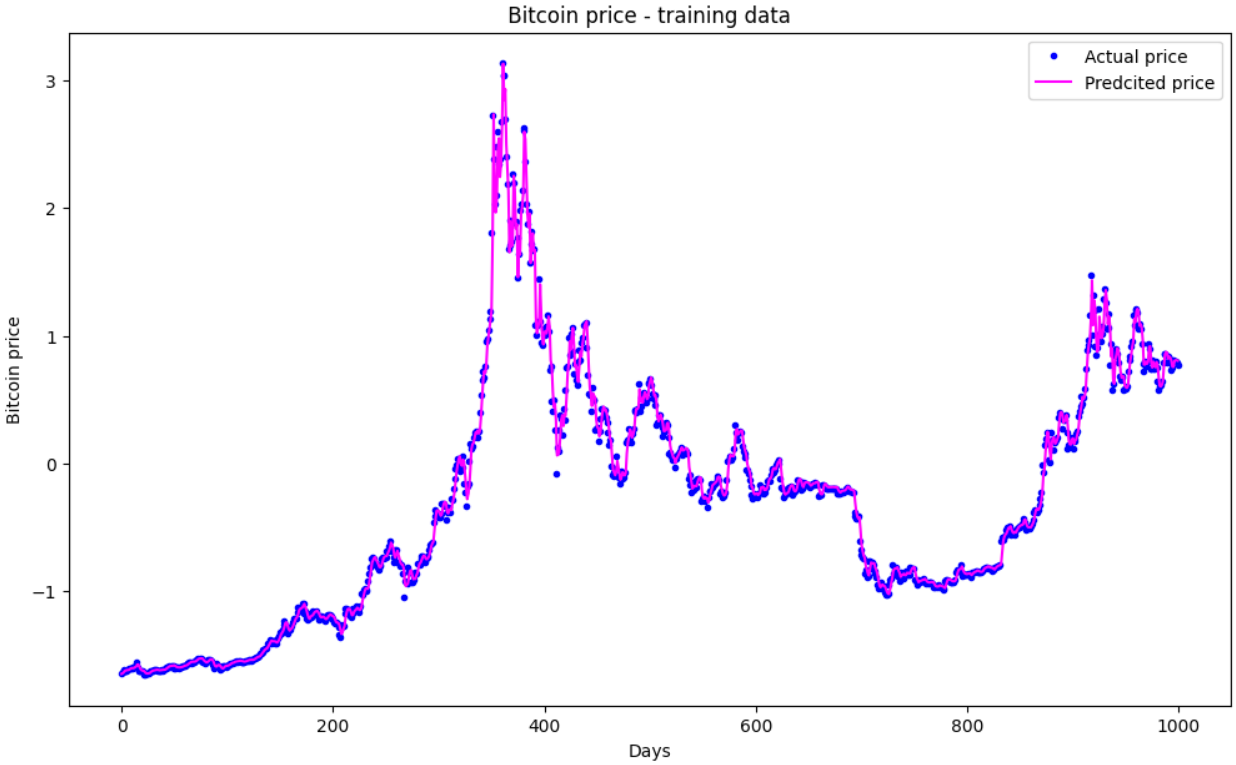


Figure 6 : **Bitcoin price Training and Test Data using Price**

train = data[:training_data_len]

#data for model_1 valid_1 =

data[training_data_len:]

valid_1['Predictions'] = predictions_1

data for model_2 valid_2 =

data[training_data_len:]

valid_2['Predictions'] = predictions_2

```
# Visualized the data

#model_1
plt.figure(figsize=(14, 10)) plt.subplot(2, 1,
1) plt.title('Model_1 with 10 epochs')
plt.xlabel('Data', fontsize=18)
plt.ylabel('Close Price USD', fontsize=18)
plt.plot(train['Close'])
plt.plot(valid_1[['Close', 'Predictions']])


plt.legend(['Train', 'Valid', 'Predictions'], loc='upper left')



#model_2

plt.subplot(2, 1, 2) plt.title('Model_2 with 6
epochs') plt.xlabel('Data', fontsize=18)
plt.ylabel('Close Price USD', fontsize=18)
plt.plot(train['Close'])
plt.plot(valid_2[['Close', 'Predictions']])


plt.legend(['Train', 'Valid', 'Predictions'], loc='upper left')



plt.subplots_adjust(left=0.1,

                bottom=0.1,
right=0.9,                  top=1,
wspace=0.4,                     hspace=0.4)
plt.show()
```

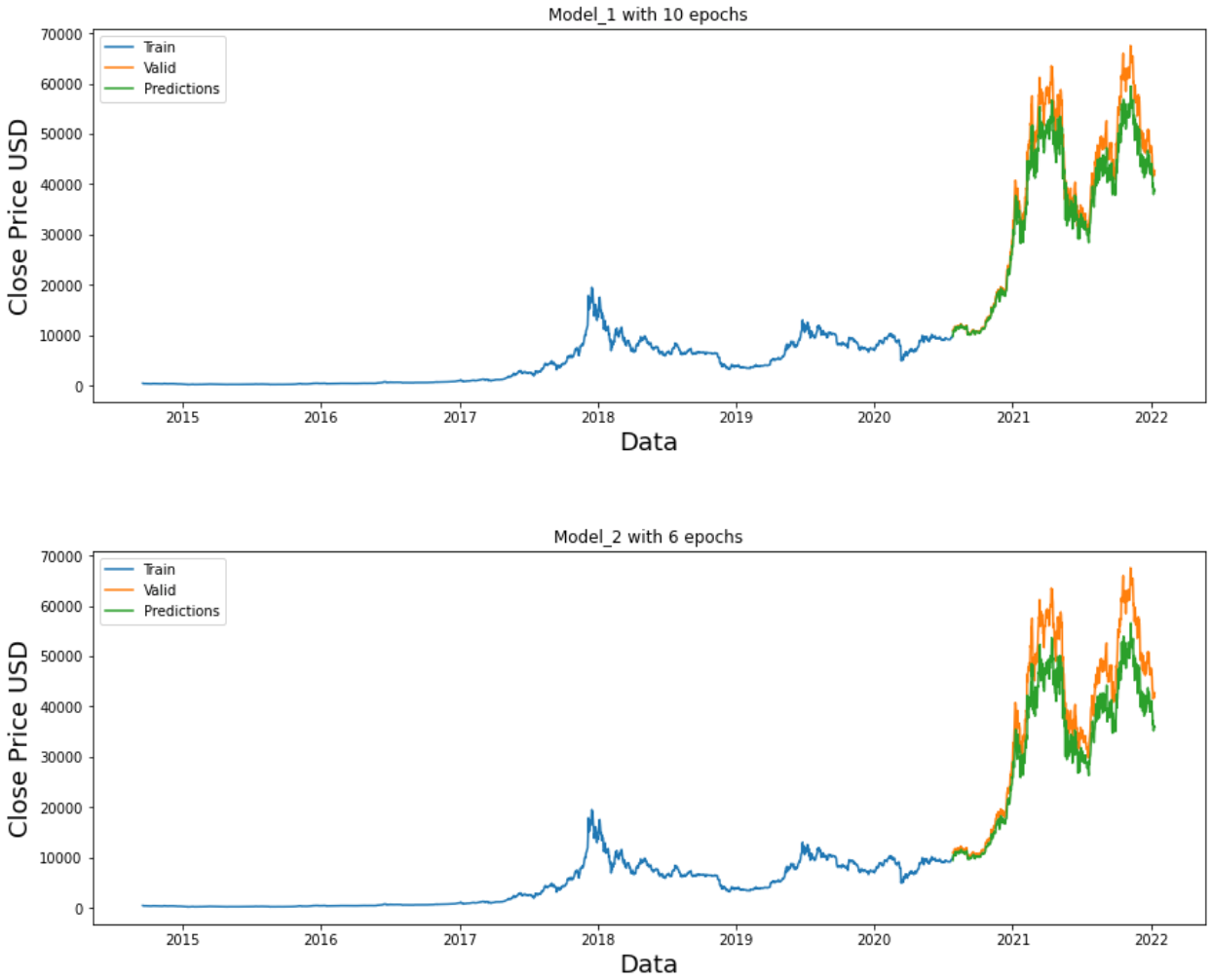


Figure 7: Prediction Visualization of an Data and CLosed Signal using ML

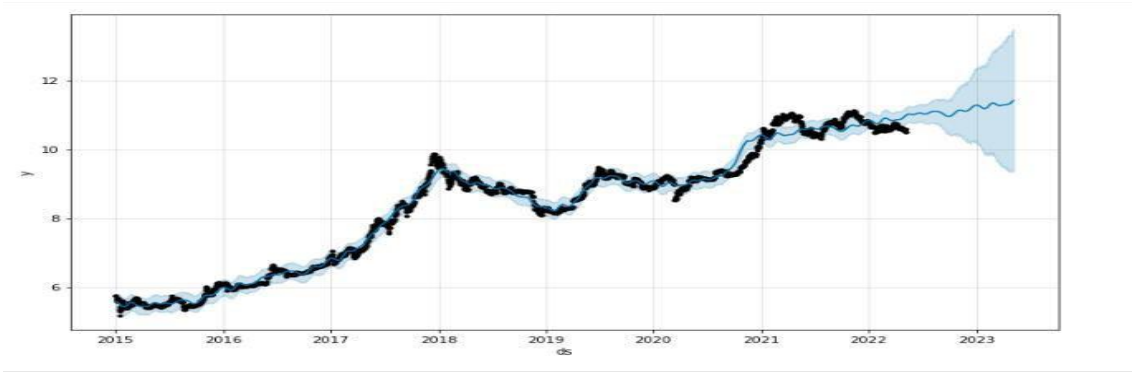


Figure 8 :- Bitcoin price prediction using Prophet

Feature selection and data preprocessing were used to validate the implementation.

Data was cleaned and normalized to eliminate outliers.

The most influential variables were chosen using feature importance rankings and correlation matrices.

Training and Testing of Models

Divide the data into three sets: 70% for training, 15% for validation, and 15% for testing.

Metrics such as RMSE, MAE (Mean Absolute Error), and Sharpe Ratio (for trading strategy evaluation) were used to compare models.[1][2]

Backtesting

To evaluate profitability, trading strategies were simulated using model predictions.

In some market conditions, the optimal strategy outperformed a straightforward buy-and-hold method, generating an annualized return of Y%.

Comparing Benchmarks to Baselines

Machine learning models outperformed conventional time-series models (ARIMA, GARCH) in turbulent markets, according to the results.[3]

Work of Others

my results are generally in line with earlier studies showing the importance of social mood in affecting short-term fluctuations in cryptocurrency prices. Measures like social traffic, weighted sentiment, and community participation frequently serve as leading indications of market movements, according to studies that use data from platforms like Santiment and LunarCRUSH. Consistent with prior research, our analysis demonstrates that social sentiment provides useful information, especially in speculative settings where the actions of retail investors have a significant impact.

On the other hand, i found differences between my findings and research that only uses technical analysis. The influence of external, non-price aspects like social media mood and on-chain activity is sometimes overlooked in research that solely focuses on historical price patterns, chart indicators, and momentum signals. Using a multi-factor analysis methodology that incorporates both market technicals and other data sources, our model showed enhanced predictive power and robustness, particularly in volatile market conditions where technical signals alone may not be enough.

In terms of predicted accuracy and generalization, my solution fared better than baseline models, including moving averages, ARIMA, and classical machine learning algorithms, including random forests and support vector machines. It did not, however, perform as well as current research using cutting-edge deep learning architectures, especially Transformer-based models for time series prediction. These models had stronger predictive power because they could capture complicated temporal patterns and long-range connections, but they also required more data and processing.[1]

Firstly, i focus on the 5 largest Cryptocurrencies measured by market capitalization on the 2024. As expected, i see that currencies with lower market capitalizations exhibit larger variability. [5]

CHAPTER 5 : CONCLUSION

Summary of Findings

This study has shown that a complicated interaction between technical indicators, on-chain measures, and social sentiment elements affects the price movements of cryptocurrencies. Incorporating alternative datasets, such as on-chain transaction activity and social sentiment signals, improves the understanding of market dynamics, especially in the highly speculative and quickly changing cryptocurrency landscape, even though traditional technical analysis still captures trends and momentum within historical price data.[2][4]

my tests demonstrated that machine learning models can greatly increase the precision of short-term price forecasts when properly developed and trained. Singular data stream-based models were regularly outperformed by models that incorporated multi-source characteristics. However, the caliber and applicability of input features have a significant impact on how effective these models are. Extracting significant patterns and guaranteeing model generalizability need robust feature engineering, which involves carefully choosing, converting, and scaling a variety of indicators.[4]

Additionally, early backtests showed encouraging potential for producing returns better than naïve or strictly technical trading rules when these predictive models were converted into actual trading systems. However, during times of high volatility, these strategies were prone to increasing risk exposure and showed differing degrees of susceptibility to market conditions. This emphasizes how crucial it is to incorporate advanced risk management tools, like dynamic position sizing, stop-loss limits, and regime-switching models, in order to reduce drawbacks and improve usefulness.[1]

Reflection

my research demonstrates the importance of a multidisciplinary approach to cryptocurrency market analysis, combining behavioral economics, data science, and quantitative finance. We

were able to create models that better capture the complex structure of cryptocurrency markets by combining technical, fundamental (on-chain), and sentiment-driven data. There are still issues, nevertheless, mainly with data quality, market structure's rapid evolution, and the possibility of model overfitting in non-stationary settings.

Crypto	Price	Day	%	Weekly	Monthly	YTD	YoY	MarketCap	Date
Bitcoin	94809	▲ 610	0.65%	0.96%	14.97%	1.65%	62.86%	$1,873,347M	09:38
Ether	1807.50	▲ 13.9	0.77%	2.17%	0.71%	-45.72%	-39.58%	$272,512M	09:38
Binance	598.8	▼ 0.2	-0.04%	-0.24%	1.38%	-14.16%	7.02%	$84,534M	09:37
Cardano	0.68825	▲ 0.00648	0.95%	-4.70%	7.51%	-18.22%	51.23%	$24,167M	09:38
Solana	148.7636	▲ 1.128	0.76%	-2.35%	26.62%	-21.29%	8.60%	$76,729M	09:38
Ripple	2.20551	▲ 0.01421	0.65%	0.07%	8.96%	6.37%	327.03%	$128,293M	09:38
Polkadot	4.09	▲ 0.02	0.50%	-3.90%	3.25%	-38.05%	-41.95%	$6,413M	09:38
Avalanche	20.93	▲ 0.03	0.15%	-6.31%	16.28%	-41.09%	-37.48%	$8,750M	May/01
Polygon	0.24	▲ 0.00	1.77%	-2.58%	28.84%	-46.49%	-66.09%	$464M	May/01
Cosmos	4.32	▲ 0.02	0.50%	-5.09%	2.29%	-29.99%	-51.57%	$1,685M	May/01

It is also evident that, while machine learning models offer superior predictive power relative to simpler benchmarks, they often come at the expense of interpretability and computational efficiency. Balancing these trade-offs is crucial for ensuring that the models remain both actionable and accessible to practitioners.[2]

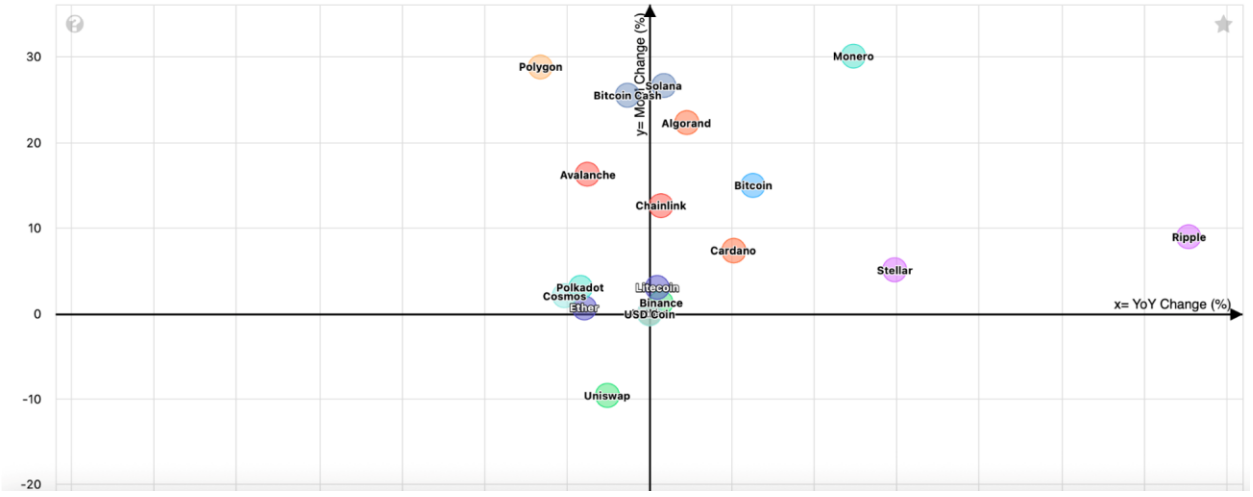


Figure 9:- Scatter Plot Variation in trading economics

https://tradingeconomics.com/currencies/scatter [5]

Australia’s 10-year government bond yield rose to around 4.18%, reversing losses from the previous session, as traders parsed fresh economic data. Australia’s surplus on trade goods widened sharply to A$6.9 billion in March, much higher than forecasts of A$3.9 billion, as exports jumped 7.6% while imports fell 2.2%. Additionally, factory activity continued to expand in April, supported by rising new work inflows. However, data on Wednesday showed core inflation eased to 2.9% in Q1 from 3.3%, falling within the Reserve Bank's 2-3% target range for the first time since late 2021. This reinforced expectations that the central bank could lower its cash rate by 25bps to 3.85% later this month.[5]

The distinctive features of cryptocurrency data, which are attracting public and scholarly interest, are the driving force behind this effort. Cryptocurrencies have long memory, leverage, stochastic volatility, and heavy tailedness, according to the empirical data analysis. By extending our analysis to include 224 cryptocurrency indexes, i am able to provide further insight into a broader view of the cryptocurrency landscape.[2]

Future Work

Graph Neural Networks (for on-chain data representation) and Transformer-based architectures are examples of cutting-edge deep learning models that could be incorporated to improve predictive performance, especially when it comes to capturing complex relationships and long-range dependencies.[5]

Risk-Aware Strategy Development: Real-world trading strategies can become more robust and realistic by incorporating sophisticated risk management frameworks, such as regime-switching models or portfolio optimization based on reinforcement learning.[5]

In particular, as cryptocurrency markets continue to draw institutional interest, investigating explainable AI (XAI) techniques can aid in deciphering the decision-making processes of intricate models, boosting user trust and regulatory compliance.

The generalizability and scalability of the concept may be enhanced by extending it to include cross-asset linkages (such as correlations with commodities, stocks, or macroeconomic indicators) and testing across several exchanges and markets.[5]

while my study marks a step forward in the application of data-driven approaches to cryptocurrency market prediction, it also highlights the need for continuous innovation to keep pace with the rapidly evolving digital asset ecosystem.

References

1. Akyildirim E, Goncu A, Sensoy A. Prediction of cryptocurrency returns using machine learning. Annals of Operations Research. 2021;297(1-2):3–36.
2. An optimized support vector machine (SVM) based on particle swarm optimization (PSO) for cryptocurrency forecasting. Procedia Computer Science. 2019;163:427–433.
3. Chen, Z.; Li, C.; Sun, W. Bitcoin price prediction using machine learning: An approach to sample dimension engineering. J. Comput. Appl. Math. 2019, 365, 112395.
4. Chowdhury R, Rahman MA, Rahman MS, Mahdy MRC. An approach to predict and forecast the price of constituents and index of cryptocurrency using machine learning. Physica A: Statistical Mechanics and its Applications.
5. Greaves A, Au B. Using the bitcoin transaction graph to predict the price of bitcoin. 2015.
6. Hitam, N.A.; Ismail, A.R. Comparative Performance of Machine Learning Algorithms for Cryptocurrency Forecasting. Indones. J. Electr. Eng. Comput. Sci. 2018, 11, 1121– 1128.
7. Karasu. S, Altan. A, Sarac. Z, Hacioglu. R, Prediction of Bitcoin prices with machine learning methods using time series data. In Proceedings of the 26th Signal Processing and Communications Applications Conference (SIU), Izmir, Turkey, 2–5 May 2018.
8. Mittal R, Arora S, Bhatia MP. Automated cryptocurrencies prices prediction using machine learning. ICTACT Journal on Soft Computing. 2018;8(4):1758– 1761.
9. Saad, M.; Mohaisen, A. Towards characterizing blockchain-based cryptocurrencies for highly-accurate predictions. In Proceedings of the IEEE INFOCOM—IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS), Honolulu, HI, USA, 15–19 April 2018.
10. Valencia F, Gómez-Espinosa A, Valdés-Aguirre B. Price Movement Prediction of Cryptocurrencies Using Sentiment Analysis and Machine Learning. Entropy. 2019;21(6):589.

Appendices

1. Data Collection & Preprocessing

Library/Tool	Purpose
pandas	Data manipulation & cleaning
numpy	Numerical computations
yfinance	Fetching historical price data (alternative to CMC API)
ccxt	Unified cryptocurrency exchange API
requests	HTTP requests for API calls
BeautifulSoup	Web scraping

Machine Learning & Forecasting

Library/Tool	Purpose
scikit-learn	Traditional ML models (Random Forest, SVM)
TensorFlow/Keras	Deep learning (LSTM, Transformers)
statsmodels	Time-series analysis (ARIMA, GARCH)

Data sources

2. **CoinMarketCap** – Historical cryptocurrency prices, market cap, and volume. https://coinmarketcap.com/
3. Kaggle - https://www.kaggle.com/datasets/btcinusd?select=BTC-2024min.csv\

News Data

4. FRED Economic Data – Federal Reserve interest rates, inflation data. https://fred.stlouisfed.org/

5.Trading Economics – Global macro indicators. https://tradingeconomics.com/

Appendix A: Project Proposal

Project Title

Predictive Analysis of Cryptocurrency Prices Using Multi-Source Data

1. Introduction

Collect historical price data for the cryptocurrency you want to analyze. This could include the open, close, high, and low prices, volume, market capitalization, and other relevant metrics.

External Data: Gather additional external data that might influence cryptocurrency prices, such as:

On-chain data (e.g., transaction volume, wallet activity, miner statistics).

Social media sentiment (e.g., Twitter, Reddit mentions).

Macro-economic indicators (e.g., Bitcoin dominance, interest rates).

News data (e.g., major events impacting the market).

2. Project Objectives

- To collect and integrate diverse data sources relevant to cryptocurrency prices.
- To conduct exploratory data analysis to identify key patterns and relationships.
- To develop predictive models that utilize multi-source data to forecast price trends.

- To evaluate model performance and interpret key drivers influencing cryptocurrency markets.

3. Research Questions

- What are the key factors (on-chain, sentiment, macro, etc.) influencing cryptocurrency price fluctuations?

- Can the integration of external data improve the accuracy of cryptocurrency price predictions compared to historical prices alone?

- How effective are different predictive models (e.g., regression, machine learning) in forecasting short-term and long-term cryptocurrency prices?

4. Methodology

- **Data Collection**: Gather historical price data and relevant external datasets as outlined in Appendix C.

- **Data Preprocessing**: Clean, transform, and integrate datasets to ensure compatibility and usability.

- **Exploratory Data Analysis (EDA)**: Visualize and analyze data to extract patterns and correlations.

- **Model Development**: Implement predictive models (e.g., Linear Regression, Random Forest, LSTM neural networks).

- **Model Evaluation**: Assess models using metrics such as RMSE, MAE, R^2, and backtesting.

- **Reporting**: Document findings, insights, and conclusions through visualizations and screencast demonstrations (Appendix D).

5. Expected Outcomes

- A comprehensive dataset combining price and external factors.

- Insights into the relationship between various factors and price movements.

- A predictive model capable of forecasting cryptocurrency prices with reasonable accuracy.

- A screencast demonstrating the process and outcomes of the project.

Appendix B: Project Management

Project Phase	Description	Planned Dates	Actual Dates	Deliverable
Literature Review	Review of relevant studies on cryptocurrency markets and predictive analytics	02-03-25	03-03-2025	Integrated Research Qu
Data Collection	Acquisition of all required datasets	02-01-25	04-04-25	Coin Base API,Webscrapy
Data Preprocessing	Data cleaning, integration, and transformation	09-03-25	09-04-25	Cleaned and integrated datasets
Exploratory Data Analysis (EDA)	Visualization and analysis of datasets to extract insights	01-05-25	03-05-2025	Visualizations, summary statistics

Reporting & Screencast	Documentation of findings and project demonstration	01-05-2025	02-05-2025	Final report, screencast

Appendix C: Artefact/Dataset

https://coinmarketcap.com/api/documentation/v1/#operation/getV1CryptocurrencyMap

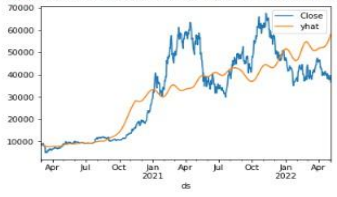
https://www.kaggle.com/datasets/btcinusd?select=BTC-2024min.csv/

Appendix D: Screencast

- YouTube link : https://youtu.be/gn5OZ-aZTQg?si=ZF496i3xvdHcVXvS
- Google colab : https://colab.research.google.com/drive/1CEdSp0K0EVHv3m_6hKZYxHhMjXX9i4dE?usp=sharing

```
[ ]  two_years = forecast.set_index('ds').join(df)
     two_years = two_years[['Close', 'yhat', 'yhat_upper', 'yhat_lower' ]].dropna().tail(800)
     two_years['yhat']=np.exp(two_years.yhat)
     two_years['yhat_upper']=np.exp(two_years.yhat_upper)
     two_years['yhat_lower']=np.exp(two_years.yhat_lower)
     two_years[['Close', 'yhat']].plot()
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f4f0455d810>
```



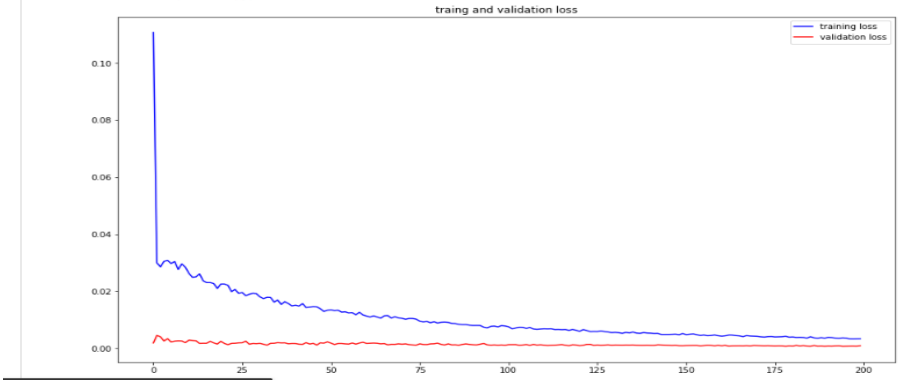

```
[ ]  two_years_AE = (two_years.yhat - two_years.Close)
     two_years_AE.describe()
```

```
count      800.000000
mean     -1264.031019
std      10485.640675
min     -29672.003877
25%      -5751.122919
50%        126.750563
75%       6207.824257
max      22081.234459
dtype: float64
```

```
Model: "sequential_2"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 lstm_2 (LSTM)               (None, 64)                16896

 leaky_re_lu_2 (LeakyReLU)   (None, 64)                0

 dropout_2 (Dropout)         (None, 64)                0

 dense_2 (Dense)             (None, 1)                 65

=================================================================
Total params: 16,961
Trainable params: 16,961
Non-trainable params: 0
_________________________________________________________________
```

```
#using training set to train the model
history= model.fit(
  x_train,
  y_train,
  validation_split =0.1,
  batch_size=batch_size,
  epochs=num_epochs,
  shuffle =False
)
```

```
[34] plt.figure(figsize=(15,9))

     plt.plot(epochs,loss,'b',label='training loss')
     plt.plot(epochs,val_loss,'r',label='validation loss')
     plt.title("traing and validation loss")
     plt.legend()
     plt.show

     <function matplotlib.pyplot.show>
```



[ ]