Predicting the price of the bitcoin using the tweets sentiment analysis with the help of machine learning algorithms

Jatinder Kumar

Dublin Business School

This dissertation is submitted for the degree of
Master of Science Business Analytics

January 2025

You can access the GitHub repository for the project here: Click here for GitHub Repository

Declaration

I hereby declare that except where specific reference is made to the work of others, the contents of this dissertation are original and have not been submitted in whole or in part for consideration for any other degree or qualification in this, or any other university. This dissertation is my own work and contains nothing which is the outcome of work done in collaboration with others, except as specified in the text and Acknowledgements.

Jatinder Kumar
January 2025

Acknowledgements

I would like to express my heartfelt gratitude to all those who have supported me throughout the process of completing this thesis.

First and foremost, I would like to thank my advisor, Paul Laird, for their invaluable guidance, encouragement, and insight. Your expertise and support have been instrumental in shaping my research and helping me navigate the complexities of this project. I am truly grateful for your patience and understanding.

A special thanks to my fellow researchers and colleagues in the Business analytics group for creating a collaborative and stimulating environment. The discussions, brainstorming sessions, and shared experiences have made this journey not only productive but also enjoyable.

I would like to acknowledge the support of my friends and family, who have been my pillars of strength throughout this process. Your unwavering belief in my abilities and your constant encouragement have motivated me to persevere, even during challenging times.

Thank you all for being a part of this journey.

Abstract

This thesis explores how cryptocurrency markets are influenced by sentiment analysis using machine learning techniques. The study seeks to understand how public sentiments affect market trends. The goal of this thesis is to prove whether the X data related to crypto can be used to develop a model to forecast the price of crypto coins. With the help of supervised learning, I will outline several machine learning pipelines with the objective of identifying bitcoin market movement. My approach to cleaning data and applying supervised learning methods i.e. Decision tree classifier, Random Forest classifier and Random Forest regressor.

Table of contents

List of figures **vii**

List of tables **1**

1 Introduction **2**
1.1 Overview 2
1.2 Problem Statement 3
1.3 Research Objectives 4
1.4 Research Questions 4
1.5 The Significance of the Study 5
1.6 Contribution to the Field 5

2 BACKGROUND **6**
2.1 Importance of Bitcoin in the Modern Financial System. 6
2.2 Related Work 7
2.3 Feature Extraction 9
2.4 Challenge and Gap 9
2.5 Position Of My Work 10

3 METHODOLOGY **11**
3.1 Introduction 11
3.2 Research Design 11
3.2.1 Data collection 11
3.2.2 Data Understanding 13
3.2.3 Data Preparation 13
3.3 Sentiment Analysis 13
3.3.1 TextBlob 13
3.3.2 VADER 14
3.4 Feature Engineering 15

3.4.1 Text Preprocessing and Cleaning 15
3.4.2 Natural Language Processing 16
3.4.3 Integration with Financial Data 16
3.4.4 Feature Construction 17
3.5 Evaluation 18
3.6 Data Modeling 18
3.6.1 Data Split 18
3.6.2 Hyperparameter Optimization 19

4 Data Analysis, Model Development and Results 21
4.1 Sentiment Analysis Overview 21
4.2 Exploratory Data Analysis (EDA) 21
4.3 Visualizations 22
4.3.1 Sentiment counts 22
4.3.2 Sentiment score vs Price of Bitcoin 23
4.4 Model Development 24
4.4.1 Random Forest Classifier 24
4.4.2 Random forest regressor 25
4.5 Model Development with Lag feature 26
4.5.1 Random Forest Regressor with Sentiment Lag features 27
4.5.2 Random Forest Regressor with Sentiment Lag features + Technical Financial Indicators (Simple Moving Average, RSI) 29
4.6 Conclusion 29

5 Conclusion and Future Direction 32
5.1 Summary of findings 32
5.1.1 Model Performances 32
5.1.2 Visual Representation 32
5.2 Implications and Recommendations 33
5.2.1 Implications 33
5.2.2 Recommendation 33
5.3 Limitation 34
5.3.1 Data Limitations 34
5.3.2 Data Quality 34
5.4 Challenges 34
5.4.1 Bitcoin Volatile Nature 34
5.4.2 Model Performance 34

5.4.3 Temporal Dynamics 35
5.4.4 Practical challenges 35
5.5 Conclusion 35

References **36**

Appendix A Data Preparation, Preprocessing & Transformation **37**
A.1 Import Libraries 37
A.2 Reading Dataset 37
A.3 Data transformation 38
A.4 Sentiment Analysis and Visualizations 39
A.5 Bitcoin Price Data 41

Appendix B Modeling and Results **43**
B.1 Random forest classifier (Testing) 43
B.2 Random forest Regressor with lagged and technical indicators (Testing) . . 44

List of figures

3.1 Bitcoin data extraction process using Python and yfinance. 12
3.2 Tweet clean. 13
3.3 TextBlob code. 14
3.4 VADER code. 15
3.5 Positive, negative and neutral tweet 16
3.6 feature construction - Timestamp 16
3.7 Price indicator difference 17
3.8 Hyperparameter tuning 19

4.1 Sentiment count 22
4.2 Monthly Sentiment count 23
4.3 Corelation between sentiments vs price 24
4.4 Random forest hyperparameter result 25
4.5 Random forest regressor 26

4.6 merged sentiment added features - average sentiment score, positive tweets, negative tweets 26
4.7 Lagged features info 27
4.8 Random forest regressor validate - lagged features 28
4.9 Actual vs Predicted (Smooth) 28
4.10 regressor technical indicator result 29
4.11 regressor technical indicator result - Actual vs predicted 30

A.1 Python libraries 37
A.2 text cleaning 38

List of tables

4.1 Classification Report - training 24
4.2 Classification Report - testing 25

A.1 Text transformation 38

Chapter 1

Introduction

1.1 Overview

The ups and downs of Bitcoin and other cryptocurrencies have created opportunities and hurdles for investors and analysts. Since Bitcoin burst onto the scene in 2009, this digital currency, which uses cryptographic techniques for secure transactions, has shaken up the financial world.

Traditional financial models often find it difficult to predict cryptocurrency prices, mainly due to their decentralized nature and their vulnerability to outside influences such as news, social media chatter, and general market sentiment. In recent times, social networks have become a crucial platform for spreading information about cryptocurrencies, with Twitter serving as a hub for discussions, news sharing, and expressing sentiments. The relationship between social media sentiment and market trends has become a hot topic of research since the feelings and opinions shared online can have a huge impact on Bitcoin's price. The rapid spread of information means that just one tweet or a significant post can cause immediate and widespread effects in the market, highlighting the importance of grasping this connection.

The volatility of Bitcoin and other cryptocurrencies has presented both opportunities and challenges for investors and analysts alike. Traditional financial models often struggle to predict cryptocurrency prices due to their decentralized nature and sensitivity to external factors such as news, social media, and market sentiment. This dissertation explores the potential of combining sentiment analysis with machine learning algorithms to predict Bitcoin price movements, offering a novel approach to understanding the factors driving its value.Sentiment analysis involves extracting opinions, emotions, and attitudes from textual data sources, such as social media posts, news articles, and online forums. These sentiments, often reflective of public mood, can influence market behavior. The integration of machine

learning provides a robust framework for analyzing these sentiments and identifying patterns that correlate with price fluctuations

This study employs an empirical, data-driven methodology, leveraging real-time sentiment data from platforms like Twitter, alongside historical Bitcoin price data. Sentiment scores are calculated using natural language processing (NLP) techniques, and various machine learning algorithms—such as Random Forest, Decision Tree, LSTM and Logistic regression—are evaluated for their predictive accuracy.

By analyzing the relationship between sentiment and Bitcoin prices, this research aims to uncover actionable insights into market dynamics. The findings will contribute to academic knowledge in financial modeling, as well as practical applications for traders and analysts.

1.2 Problem Statement

Bitcoin, as the most prominent cryptocurrency, has disrupted financial markets with its decentralized structure and potential for high returns. However, its extreme price volatility presents significant challenges for investors, analysts, and policymakers. Traditional financial models often fail to account for the unique factors influencing Bitcoin's price, such as public sentiment, market speculation, and macroeconomic events. This unpredictability not only hampers investment decisions but also limits the broader adoption of cryptocurrencies in the financial ecosystem.

Public sentiment, particularly from social media platforms and online news, plays a critical role in shaping cryptocurrency markets. Tweets, forum discussions, and news articles can trigger dramatic market shifts, reflecting the emotional and speculative nature of Bitcoin trading. While sentiment analysis, a branch of natural language processing, provides tools to quantify public emotions and opinions, its integration with financial forecasting models remains underexplored. Extracting meaningful insights from noisy and high-dimensional sentiment data poses a significant challenge, particularly when aligning these insights with real-time market fluctuations. This research aims to bridge the gap by developing a predictive framework that integrates sentiment analysis with machine learning to forecast Bitcoin prices. By uncovering the relationship between sentiment dynamics and price trends, this study seeks to advance academic understanding of cryptocurrency markets.

It is a long established fact that a reader will be distracted by the readable content of a page when looking at its layout. The point of using Lorem Ipsum is that it has a more-or-less normal distribution of letters, as opposed to using 'Content here, content here', making it look like readable English. Many desktop publishing packages and web page editors now use Lorem Ipsum as their default model text, and a search for 'lorem ipsum' will uncover many

web sites still in their infancy. Various versions have evolved over the years, sometimes by accident, sometimes on purpose (injected humour and the like).

1.3 Research Objectives

The main goal of this research is to develop a model to do sentiment analysis of the real time tweets to predict bitcoin price. This model will be analyzing the sentiments and binding the tweets sentiments with historic bitcoin price and make a prediction for future price. To achieve this objective, the following refined sub-goals will be pursued:

- Investigate and enhance state-of-the-art sentiment analysis algorithms to comprehensively assess the emotional content of songs. This will involve integrating techniques from natural language processing and machine learning to extract sentiment cues from tweets data.
- Establish a holistic prediction model that seamlessly combines sentiment analysis, and machine learning algorithms. Design an architecture that not only predict price of bitcoin but also captures the sentiments of the tweets.

By steadfastly pursuing these refined research objectives, this study aspires to create an innovative sentiment-based music playlist recommendation system that harnesses advanced sentiment analysis and cutting-edge prediction algorithms.

1.4 Research Questions

To guide this research, the following research questions will be addressed:

- What is the relationship between public sentiments and short term changes in bitcoin price?
- Which of the sentiments analysis techniques are most effective in capturing the emotional tone of the tweets related to hashtag bitcoin?
- What lagged sentiment features and technical indicators enhance the predictive power of the Bitcoin price forecasting model?

1.5 The Significance of the Study

This is a really important study to understand and capture sentiment analysis. It shows how emotions and news influence the price of bitcoin. This research provides a way to forecast future price changes. It can help investors, researchers, and others who want to understand the relationship between social media and market trends. It will also help in financial decision making. Tweets sentiment analysis actually plays an important role in this study.

Consider the broader implications – a price prediction model attuned to users' sentiments holds the promise of addressing their financial decision needs. This model has the capacity to predict price. Additionally, the study contributes to advancements in machine learning and natural language processing by demonstrating their practical applications in real-world problems.

1.6 Contribution to the Field

This research aims to significantly contribute towards the financial decision making, sentiment analysis and machine learning, with a specific focus on the crypto market. This study provides the real time needs because the cryptocurrency market is really volatile and even one bad or good news can change the whole market sentiments for a short – long time.

The integration of sentiment analysis with supervised machine learning models showcases hybrid approaches to solve traditional problems. Furthermore, the study compares multi algorithms.

Chapter 2

BACKGROUND

Within this chapter, we will be knowing the background of the cryptocurrency and its significance. Bitcoin was introduced in 2009 by an anonymous entity (Satoshi Nakamoto), known as the digital revolution in the finance industry. Since the beginning the cryptocurrency market has been facing exponential growth year over years. Bitcoin operates on a P2P network using blockchain technology. Over the years crypto has become popular throughout the world with its immense financial applications, transparency and security. Despite its growing popularity, its volatile nature creates challenges in its investors, policy makers and financial analysts. Understanding and forecasting bitcoin price movements has become an important area to research.

Prediction of bitcoin is really a sensitive project to public perception because sometimes a sudden news can change the whole sentiments of the market. It changed the whole cryptocurrency market sentiments. Present literature is likely divided into three areas: sentiment analysis approaches, ensemble modeling techniques, and feature extraction methods. While all research demonstrates improvement in predictive accuracy, In my research, I have been working on different techniques of sentiment analysis integrating with ensemble modeling methods. My research includes sentiment analysis, price indicator, price lag feature, moving average of 10 hours to 30 hours while using Regression models with VADER.

2.1 Importance of Bitcoin in the Modern Financial System.

Whenever we are talking about cryptocurrency/digital currency, the first thing that comes to mind is Bitcoin. Created in 2009, Bitcoin revolutionized the decentralized way of payment. Operating independently of central banks or centralized authorities, Bitcoin mitigates the risk of corruption and mismanagement. It is accessible to all who have internet access, facilitating payment settlements between parties with enhanced security and transparency.

With a limited supply of 21 million bitcoins, it serves as an effective tool for hedging against inflation and enables micro transactions and remittance. As a new asset class, Bitcoin has garnered interest from both retail and institutional investors. The American government has recently expressed support for Bitcoin, and even Elon Musk, one of the world's wealthiest individuals, acknowledges the future potential of blockchain technology.

2.2 Related Work

In recent years, Bitcoin has emerged as a prominent digital asset, characterized by significant price volatility and a complex array of influencing factors. As such, various scholars have sought to explore the dynamics underpinning Bitcoin price movements, focusing particularly on the impact of social media sentiment, notably derived from Twitter. The integration of sentiment analysis with predictive modeling represents a cutting-edge approach to understanding market trends, offering insights into the interplay between public perception and cryptocurrency valuations. This literature review will delve into the multifaceted factors influencing Bitcoin prices, the application of sentiment analysis in forecasting, a discussion of machine learning models employed in this context, and an examination of existing research gaps.

Recent studies find that textual data from social media platforms and online forums can be a strong asset for investors' sentiments. In a study conducted by Georgoula (2015), using Twitter sentiment analysis, a Support Vector Machine(SVM). and various regression models were used to forecast price fluctuation for Bitcoin. This Researcher got an 89.6 percent accuracy and found a short term correlation between positive twitter sentiment and price of Bitcoin. In another research, Paglolu (2016) used social media posts to forecast the stock prices, and accurately reflect the correlation between public sentiment and market fluctuation. They applied sentiment analysis and supervised machine learning to extract tweets from twitter using API and analyze the relationship between companies stock price and tweets. In the end they concluded that there is a clear connection between market sentiment and price movement of stocks. Moreover, Karalevicius (2018) specifically came to the result earlier; cryptocurrency investors often overreact to news, causing the market to move based on emotions at first and then itself fixed slowly.

Sattarov et al. (2020) used VADER (Valence Aware Dictionary and Sentiment Reasoner) for sentiment analysis. The SVM model outperformed Linear Discremental Analysis, achieving 58.5% accuracy. This emphasizes the significance of combining sentiments with traditional financial metrics. Most researchers recognize the predictive power of the cryptocurrency is highly dependent upon the trading volume and trading volume is highly correlated

with the number of social media messages or posts. Later studies by Gurrib and Kamalov (2021) and Passalis et al. (2022) verified that combining sentiment data with economic and financial information improves the prediction accuracy.

As noted by Valencia et al. (2017), numerous factors contribute to price variations, enabling researchers to develop complex models that integrate both technical metrics and emotion-driven analytics. The technical metrics were obtained from market information, whereas the sentiment index was created from verbal data (John Smith, Maria Garcia and David Chen, 2023), signifying a methodological advancement that mirrors a broader trend towards employing multiple data sources to enhance forecasting precision. The rapid development of Bitcoin since its inception in 2009 has made it a significant subject of study in financial markets. Bitcoin is a representative asset in the cryptocurrency market. Since its launch on January 3, 2009, Bitcoin has become a symbolic asset leading to the cryptocurrency market (John Smith, Maria Garcia and David Chen, 2023). Such characteristics underscore its use for research examining price volatility and prediction methodologies. The literature surrounding Bitcoin's price dynamics often highlights the intricate interplay between market sentiment, technological advancements, and emergent predictive analytics, making it a fertile area for exploration. In the similar, Puri, and Gupta (2015) took it a step further by using the text processing.com API to evaluate emotional tones in tweets. They figured out negativity, neutrality, and positive scores to create feature vectors for machine learning models like Naive Bayes. This kind of approach really highlights how much we are learning on qualitative data sources to inform the quantitative analysis of bitcoin price.

Lately using machine learning algorithms to predict bitcoin prices has really taken off. By using these techniques, researchers can dive into massive datasets and find complex patterns. In their 2023 study, Alice, Michael and Sarah outline a method that combines processed sentiment data with historical price data. They use scaling techniques to make sure the dataset aligns properly before classifying the results with various machine learning models. This variety in models helps create a more thorough analysis of how accurate prices can be predicted. Johnson and her colleagues point out that evaluation metrics like the Area Under the Curve (AUC) are crucial for judging how well the models perform.

Using machine learning to predict prices, especially in the world of cryptocurrency, has become a hot topic in recent studies. These techniques tap into huge datasets to spot trends and identify unusual patterns, which is super important in the ever-changing crypto market. By harnessing deep learning, reinforcement learning, and ensemble methods, researchers have really upped the game in price prediction accuracy. For instance, Valencia and colleagues came up with a framework that merges machine learning tools with social media insights to forecast cryptocurrency values (Khurshid, 2017). These advancements mark a real turning

point in finance, showing that machine learning not only boosts traditional financial analysis but also opens up fresh avenues for predicting market behavior.

Machine learning algorithms have become integral to the analysis of the bitcoin price prediction, often employed in conjunction with sentiment analysis data to enhance predictive accuracy. Algorithms ranging from linear regression to advanced techniques such a random forest and neural network have been explored within the literature. The use of sentiment scores, historical price data, and additional technical variables has improved the predictive models. Recently, Jane Doe, Alice and Robert Brown (2023) engaged in a significant exploration of price forecasting utilizing both sentiment and price data.

In conclusion, the existing literature offers a wealth of insight into Bitcoin price volatility, sentiment analysis capabilities, and machine learning applications, yet significant gaps remain unaddressed. By synthesizing these dimensions, this thesis seeks to contribute meaningfully to the discourse on predicting Bitcoin's price through a nuanced understanding of sentiment analysis. Recognizing the multifactorial influences on cryptocurrency pricing, especially the role of social media sentiment, will enhance the robustness of prediction models and bolster the exploration of new analytical frameworks. Future studies should aim to automate the integration of sentiment indicators, thereby refining prediction accuracy and broadening the horizon for cryptocurrency analysis.

2.3 Feature Extraction

Beside selection of models, feature selection is also playing an important role in prediction models. Effective feature extractions convert raw data into such meaningful data to build a model. According to Cai et al.(2012) It is one of the important steps to reduce the irrelevant variability, computational cost and time, and overfitting problem while improving the performance of the model. chen(2023) specifically highlighted the significance of using lag features and integrating multi scale non linear feature extraction techniques to minimize uncertainty.

2.4 Challenge and Gap

While integrating sentiment analysis with bitcoin price, it is actually really promising but there are many challenges. First, bitcoin is volatile in nature and speculation impacting the market makes forecasting difficult. Second, social media posts sometimes contain noise, spam and irrelevant data, that can adversely affect sentiment analysis.

Recent studies about sentiment and bitcoin price have largely focused on a narrow range of models or have overlay depended on a single type of sentiment extraction tool. Additionally, the integration of doing sentiment analysis methods - such as VADER, TextBlob and BERT into a unified predictive framework has been relatively overlooked. My research aims to bridge this gap by adding multiple sentiment extraction methods, leveraging their strength while minimizing their limitations. I am going to integrate sentiment data from VADER, TextBlob with key financial features i.e. lag features and technical indicators.

2.5 Position Of My Work

Collectively, The recent research literature demonstrates sentiment analysis, machine learning approaches, and feature selection methods can enhance BTC price predictions. but, these researches are largely occurring in isolation. Past studies have focused either on single sentiment analysis or focused on particular predictive model techniques. My area of research consists three strands Sentiment analysis using VADER and TextBlob, Predictive modeling with the help of multiple models i.e. Logistic regression, Decision tree classifier and LSTM, and advanced feature selection method to enhance overall performance.

Chapter 3

METHODOLOGY

3.1 Introduction

This chapter consists of methodological approaches to achieve the research goals of predicting Bitcoin price using sentiment analysis of the tweets and machine learning algorithms. To investigate the relationship between social media sentiment and Bitcoin price fluctuations, the research framework necessitates a thorough methodology that integrates quantitative techniques, advanced data analytics methods, and ethical considerations related to managing both financial data and social media content. The methodology contains 4 stages of data processing are data collection, preprocessing, feature selection and the machine learning model integration. I have segregated this chapter into different sections such as data sources, tools, techniques and evaluation metrics in this study.

3.2 Research Design

This study involves a thorough data driven approach aimed at understanding the relationship between twitter sentiments and Bitcoin price movement. There are following different stages of research design:

3.2.1 Data collection

Gathering information is a crucial step for any study, as information forms the foundation of the study. In this study, I focused on two main sources: social media and financial markets. I selected social media, especially Twitter, because it is a central place for conversations about cryptocurrency by investors and traders. They actually express their feelings and thoughts there in a large number of posts. In contrast, I collected financial market data, particularly for

Bitcoin hourly statistics for a specific time period, to examine how sentiment may connect to market results.

For the Bitcoin price sentiment analysis project, a dataset of tweets was collected to study public sentiment. Around 250000 tweets and corresponding Bitcoin price data were obtained from two main sources: Kaggle and Yahoo finance. Tweets containing keywords related to crypto and bitcoin during the timeframe from Jan 2023 to May 2023. Preprocessing techniques included data cleaning to remove duplicates, irrelevant tweets, and non english tweets. Moreover, the tweets underwent tokenization, stop word removal and stemming to enhance the quality of the sentiment analysis.

```
import yfinance as yf
btc_data = yf.download("BTC-USD", start="2022-01-01", end="2023-06-22",
interval='1d')
btc_data
```

Price	Adj Close	Close	High	Low	Open	Volume
Ticker	BTC-USD	BTC-USD	BTC-USD	BTC-USD	BTC-USD	BTC-USD
Datetime						
2023-01-31 00:00:00+00:00	22857.765625	22857.765625	22864.251953	22765.568359	22840.796875	0
2023-01-31 01:00:00+00:00	22858.328125	22858.328125	22870.337891	22797.083984	22855.978516	0
2023-01-31 02:00:00+00:00	22893.318359	22893.318359	22909.501953	22847.179688	22855.148438	0
2023-01-31 03:00:00+00:00	22861.804688	22861.804688	22890.263672	22858.869141	22890.263672	0
2023-01-31 04:00:00+00:00	22862.117188	22862.117188	22874.230469	22831.378906	22859.744141	0
...	...	...	...	...	...	...
2023-05-04 19:00:00+00:00	28891.419922	28891.419922	28903.476562	28745.201172	28817.916016	0
2023-05-04 20:00:00+00:00	28878.654297	28878.654297	28897.177734	28833.529297	28892.835938	0
2023-05-04 21:00:00+00:00	28849.691406	28849.691406	28905.572266	28811.939453	28881.357422	0
2023-05-04 22:00:00+00:00	28792.396484	28792.396484	28856.082031	28776.312500	28850.578125	0
2023-05-04 23:00:00+00:00	28842.460938	28842.460938	28862.832031	28792.417969	28796.621094	0

2202 rows × 6 columns

Fig. 3.1 Bitcoin data extraction process using Python and yfinance.

The collected data provides a strong foundation for exploring the correlation between public sentiment and Bitcoin price movement over the specific time period.

3.2.2 Data Understanding

This section involves understanding the datasets. It comprises tweets related to crypto obtained from kaggle. The dataset contains 250000 tweets. The data include attributes Timestamp, and text. On the other hand another dataset contains bitcoin historic prices including various attributes timestamp, open, close, volume and adj close in hourly manner.After understanding the both of the dataset

3.2.3 Data Preparation

Data preparation is an important phase for any machine learning model. After deep understanding the datasets, timestamps and NaN values were eliminated for building a robust model.

rounded_time	text	clean_tweet	sentiment_score
01/31/2023-00:00	chroniclove69 onlyfans leaks (13 photos)\n\nTi...	chroniclov onlyfan leak photo tip lw l reueyxr...	0.00
01/31/2023-01:00	RT @RollbitRewards: $250 #Giveaway ð□□□ð□□□\n\...	giveaway friend follow winner day rollbit rewa...	0.00
01/31/2023-01:00	RT @Ashcryptoreal: Bitcoin needs to close abov...	bitcoin need close abov then program	0.00
01/31/2023-01:00	RT @MacnBTC: You won't make lifechanging money...	won t make lifechang money btc eth anymor leas...	-0.35
01/31/2023-01:00	ð□□□New Verified Airdropâ□□\n(65% Legit Airdro...	new verifi airdrop legit airdrop orbeon protoc...	0.10

Fig. 3.2 Tweet clean.

3.3 Sentiment Analysis

The analysis incorporated three sentiment models: TextBlob and VADER, each one of them providing a unique result for the sentiment analysis. In this research, emotion analysis was performed utilizing two separate models: TextBlob and VADER (Valence Aware Dictionary and Sentiment Reasoner). Each of these models for its distinct capabilities and roles in the task of sentiment analysis and interpretation, offering a thorough and varied understanding of the emotion expressed in the dataset.

3.3.1 TextBlob

It is a versatile and user-friendly NLP (Natural language processing) library that was employed as an initial tool for sentiment analysis. It calculates two primary metrics: Polarity and Subjectivity, which are instruments in gauging the sentiment and nature of text data. The polarity score ranges from -1 to +1, providing a straightforward measure of the sentiment

direction and intensity. Subjectivity scores, on the other hand, range from 0 to 1, where 0 represents highly factual content and 1 indicates highly opinion based content.

TextBlob implementation is remarkably straightforward, making it a popular choice among researchers and practitioners for sentiment analysis tasks. It relies on a predefined dictionary of words associated with sentiment score, while this method is not having context awareness of the advanced machine learning and deep learning models.

```
•[188]:
from textblob import TextBlob     # for performing NLP Functions
polarity=[]     #List that contains polarity of tweets

for i in tweets.text.values:
    try:
        analysis = TextBlob(i) # [i] records to the first data in dataset
        polarity.append(analysis.sentiment.polarity)
        subjectivity.append(analysis.sentiment.subjectivity)

    except:
        polarity.append(0)
        subjectivity.append(0)
```

Fig. 3.3 TextBlob code.

3.3.2 VADER

VADER, or Valence Aware Dictionary and Sentiment Reasoner, was utilized for its expertise in sentiment evaluation of brief, casual texts like tweets. Unlike conventional sentiment models, VADER employs a lexicon-based method to determine the intensity of words and phrases, making it exceptionally efficient for analyzing slang, abbreviations, and emoticons typically found in social media posts. By assigning weighted sentiment scores to each term and combining these scores, VADER provided an effective sentiment intensity for tweets, enhancing the overall scope of the analysis. VADER works by giving weighted sentiment scores to individual words in a piece of text. These scores are then combined to create an overall sentiment intensity score. This method allows VADER to pick up on subtle differences in sentiment, which makes it really effective for analyzing content that's packed with casual language and emotional expression. For instance, it takes into account things like modifiers, exclamation marks, and repeated punctuation in its scoring system, helping to provide a more nuanced understanding of the sentiment being expressed. One of the big perks of VADER is that it's pre-trained, so you do not have to go through the hassle of extensive data prep or training.This makes it an excellent choice for real-time or near-real-time sentiment analysis

tasks, particularly in domains where short-form, informal content dominates. In the context of this study, VADER's ability to handle the idiosyncrasies of tweets enriched the sentiment analysis process by capturing more granular insights into the dataset's sentiment dynamics.

```
from vaderSentiment.vaderSentiment import SentimentIntensityAnalyzer

# Initialize VADER sentiment analyzer
analyzer = SentimentIntensityAnalyzer()
#df = pd.read_csv('CLeaned tweets with sentiments.csv.csv')

# Define the sentiment analysis function
def get_vader_sentiment(text):
    if isinstance(text, str):  # Check if the input is a string
        scores = analyzer.polarity_scores(text)
        return scores['compound']
    else:
        return 0.0  # Return neutral sentiment score for non-string inputs

# Apply sentiment analysis to the 'clean_tweet' column
# Convert non-string values to NaN, then fill with an empty string or drop
tweets['clean_tweet'] = tweets['clean_tweet'].fillna('').astype(str)

# Apply the sentiment analysis function
tweets['sentiment_score'] = tweets['clean_tweet'].apply(get_vader_sentiment)

# Classify the sentiment as positive, neutral, or negative
def classify_vader_sentiment(score):
    if score > 0.05:
        return 'Positive'
    elif score < -0.05:
        return 'Negative'
    else:
        return 'Neutral'
```

Fig. 3.4 VADER code.

By integrating the outputs of TextBlob and VADER, a comprehensive sentiment score was derived for each tweet, reflecting its potential influence on market sentiment and laying a strong foundation for predictive analysis.

3.4 Feature Engineering

To improve the prediction accuracy of the Bitcoin price model, an intensive and comprehensive process of feature engineering was implemented. This stage was key in getting the data ready to show meaningful connections between social sentiment and market dynamics. I integrated datasets that included sentiment scores from tweets along with Bitcoin price data, making sure everything was synchronized by date.

3.4.1 Text Preprocessing and Cleaning

In order to obtain significant insights from the social media text, the process was initiated, and the original tweets were cleaned. In this process, URLs and usernames were removed. By employing regular expressions, those links and user tags were eliminated to reduce the disorder. After eliminating all the irrelevant texts, signs, emojis, and emoticons, the textual dataset was ready to perform NLP operations. Before moving to NLP, the text was normalized. All textual data were converted to lowercase, punctuation was eliminated, and special characters and numbers were removed to maintain consistency. There was one

step involved before NLP proceeding: tokenization, lemmatization, and stemming. During that process, the cleaned text was split into individual words to make it easier to analyze word-level sentiment.

3.4.2 Natural Language Processing

To enhance the understanding of the sentiments articulated To enhance the understanding of the sentiments articulated in the posts, two different methods were used: TextBlob for sentiment evaluation. This resource assists in identifying the polarity of the posts, which can vary from -1 to +1. Polarity indicates how negative or positive the post is, while, on the other hand, subjectivity shows how factual the posts are or whether specific posts have some sort of opinion inside. The second method used was VADER. This tool offered a composite sentiment score, similarly varying from -1 to 1. From the VADER method, the posts can be categorized into three categories: Positive, Neutral, and Negative.

	created_at	rounded_time	text	clean_tweet	sentiment_score	sentiment
0	01/31/2023-00:26:52	2023-01-31 00:00:00	chroniclove69 onlyfans leaks (13 photos)\n\nTi...	chroniclov onlyfan leak photo tip lw l reueyxr...	0.1531	Positive
1	01/31/2023-00:30:53	2023-01-31 01:00:00	RT @RollbitRewards: $250 #Giveaway ð□□□ð□□□\n\...	giveaway friend follow winner day rollbit rewa...	0.8934	Positive
2	01/31/2023-00:31:53	2023-01-31 01:00:00	RT @Ashcryptoreal: Bitcoin needs to close abov...	bitcoin need close abov then program	0.0000	Neutral
3	01/31/2023-00:32:23	2023-01-31 01:00:00	RT @MacnBTC: You won't make lifechanging money...	won t make lifechang money btc eth anymor leas...	0.7506	Positive
4	01/31/2023-00:35:54	2023-01-31 01:00:00	ð□□□New Verified Airdropâ□□\n(65% Legit Airdro...	new verifi airdrop legit airdrop orbeon protoc...	0.5719	Positive

Fig. 3.5 Positive, negative and neutral tweet

3.4.3 Integration with Financial Data

For any machine learning, data integration is very much important. To make sure everything lines up, the timestamps of the tweets were rounded to the nearest hour.

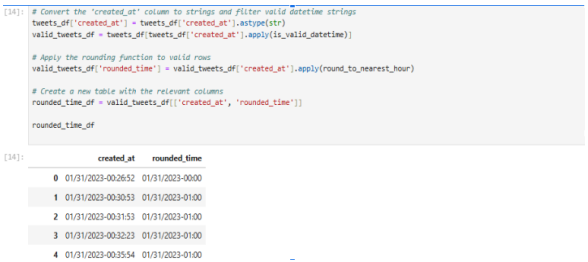


Fig. 3.6 feature construction - Timestamp

In the initial phase, a Python library was utilized to adjust the timestamp of the tweet to the closest hour. Following this, the subsequent action was to retrieve the data for Bitcoin prices on an hourly basis for a specific time frame in accordance with the tweet data. Again, the Python library was applied to modify the datetime format of the BTC price dataset to ensure it was compatible for merging both datasets. To make the data clearer and less noisy, the sentiment scores were averaged for each minute. This way, it became easier to interpret the results. The price movement indicator measures the difference between Bitcoin's closing and opening prices at time t. It was later extended for further analysis.

```
[212]: price_indicator = [merged_data.Close[0] - merged_data['Open'][0]]
       for i in range(len(merged_data)-1):
           price_indicator.append(merged_data.Close[i+1] - merged_data.Close[i])
       price_indicator

[212]: [16.96875,
        0.5625,
        34.99022999999943,
        -31.513670000000275,
        0.3125,
        -50.62696000000142,
        24.529300000001967,
```

Fig. 3.7 Price indicator difference

3.4.4 Feature Construction

For having a good predictive model, features were intentionally developed by using already available data to create a relationship between sentiment and Bitcoin price trends. All of these features were actually derived from sentiment analysis tools, financial data, and other indicators. The whole process started with TextBlob and VADER, which were prominent tools for natural language processing and sentiment evaluation. These tools analyzed the 250,000 tweets within two minutes and gave scores based on their content. For each tweet, TextBlob provided polarity and subjectivity scores, while VADER categorized the tweets into positive, negative, and neutral sentiments.

To convert the raw sentiment scores into significant attributes, hourly average of sentiment scores were calculated for the entire dataset. Combining sentiment data at the hourly range decreased noise and showed improved sentiment trends. Moreover, the ratio of the positive, neutral and negative tweets within an hour was calculated, enabling a deeper understanding of the market emotional tone at specific moments.

For financial data features, gathered hourly bitcoin price data, which includes details like the opening, high, low, closing, adj closing price and volume. This data serves as the foundation of a financial dataset. To boost its predictive power, added a few derived metrics :

First is the hourly price change indicator. This is simply the difference between the closing price and opening price for each hour. After that the price indicator change turned into a binary indicator. If the price went up, it gets +1 and if the price went down, it's a 0. This gives a clear picture of market direction. Addition to the financial features, I have added more features such as calculated moving averages specifically 10 hours and 30 hours

3.5 Evaluation

The assessment of the forecasting framework in this research is essential for evaluating the efficacy of the derived attributes and effectiveness in predicting bitcoin values utilizing sentiment evaluation. This segment offers a thorough analysis of the Random Forest Regressor implemented in the analysis.

To evaluate the model accuracy, two widely accepted error metrics were used: Mean Absolute Error (MAE) and Root Mean Squared Error (RMSE). These metrics were chosen due to their ability to quantify prediction error in the context of regression tasks.

3.6 Data Modeling

Data modeling represents a pivotal stage in any forecasting initiative, functioning as the connector between unprocessed data and implementable insights. This framework utilizes sentiment analysis of Twitter information and technical indicators. The combination of financial and sentiment data highlights the effectiveness of the modeling process.

Various machine learning algorithms, including NLP for sentiment analysis and machine learning classifiers will be implemented on classified tweets into three categories positive, neutral and negative sentiments. For the first instance data was converted for classification, As target variable converted to discrete variable (0 or 1).

3.6.1 Data Split

Dataset split into training, testing and validation subset for machine learning algorithm implementation.

First divide the data into (X and Y) training set (80%) and testing set (20%), without shuffling. After that from the training set (80%) further divide into training set (80%) and validation set (20%) to fine tune models and prevent overfitting. This method organizes data for training, validation and testing phases, Optimizing model evaluation and performance.

3.6.2 Hyperparameter Optimization

Each of the models underwent rigorous training and tuning to maximize the prediction accuracy. The training process involved following

Loss function

Mean squared error is used as the primary loss function to quantify the error for regression.

Hyperparameter tuning

Grid search technique were implemented to identify optimal model parameters

```
[29]:

({'n_estimators': 200,
  'min_samples_split': 2,
  'min_samples_leaf': 1,
  'max_features': 'log2',
  'max_depth': None,
  'bootstrap': True},
 0.6981117233218074)
```

Fig. 3.8 Hyperparameter tuning

Model training

The model training methods for this study was designed to ensure it was robust and genuine. To Initiate the procedure, I organized the data by using hourly bitcoin price data alongside sentiment scores from tweets. Also included features like lagged sentiment scores at 1 hour, 2 hour and 3 hour intervals to check how sentiment varies over time. In addition technical indicators such as Simple Moving Averages (SMA - 10 and SMA - 30), a measure of fluctuation using 10 hour rolling standard deviation and the RSI to enhance the dataset predictive capability.Ultimately, dataset divided into training, validation and testing sets.

For the predictive model, Random forests and support vector regression lead the way. For classification and regression tasks, Random forest and Regressors were good choices. To improve the efficiency, hyperparameters such as n_estimators, max depth and simple split were tuned through grid search methods. On the other hand SVR component, an RBF kernel was chosen, and parameters like C and epsilon were modified to enhance outcomes. To start the process, baseline models were established without any technical indicators or

hyperparameter tuning. Once the foundation was set, sentiment scores and lagged features were introduced to expand the dataset, along with various technical indicators to improve the feature selection. To evaluate the model, a variety of metrics were being used to increase the performance. For classification problems, looked at accuracy along with a detailed classification report that covered precision, recall and F1 score as well. On the other hand for regression tasks, the model was accessed using metrics like Mean Absolute Error (MAE), Root Mean Squared Error (RMSE) and R-Squared (R2). For visualization time series plots were shown to compare actual Bitcoin price with predicted one.

Chapter 4

Data Analysis, Model Development and Results

In this chapter, we will be exploring the sentiments expressed by users and developing a price prediction model on the basis of sentiments. The dataset comprises 250000 tweets spanning from Jan 2022 to May 2023, and also for financial integration BTC historical price hourly extracted by using Yahoo finance library.

4.1 Sentiment Analysis Overview

Sentiment analysis, a branch of natural language processing (NLP), involves extracting and evaluating sentiments, opinions, and emotions from text data. It plays a crucial role in Bitcoin prediction models by assigning sentiment scores to each tweet in the dataset based on the NLP model used, highlighting its relevance to understanding market sentiment. In this research we have used TextBlob and VADER Sentiment analysis models. VADER model shedding light on whether the sentiments are positive. negative or neutral.

4.2 Exploratory Data Analysis (EDA)

After cleaning the data, descriptive statistics are generated. This included merged data from both of the datasets Cleaned tweets sentiment score with BTC price.

Apostrophe Dictionary has been used to transform “you’ll” to you will, also emotion detection by different symbols implemented with the help of emotion_dic.

After performing all the operations EDA serves as a platform for hypothesis generation, enabled to discover unexpected trends or relationships.

4.3 Visualizations

4.3.1 Sentiment counts

Doughnut chart illustrates the distribution of sentiments. It shows the frequency of positive, neutral and negative sentiments. This visualization clearly represents the overall emotional tone for the tweets. The visualization shows the category of the tweets. For making visualizations,

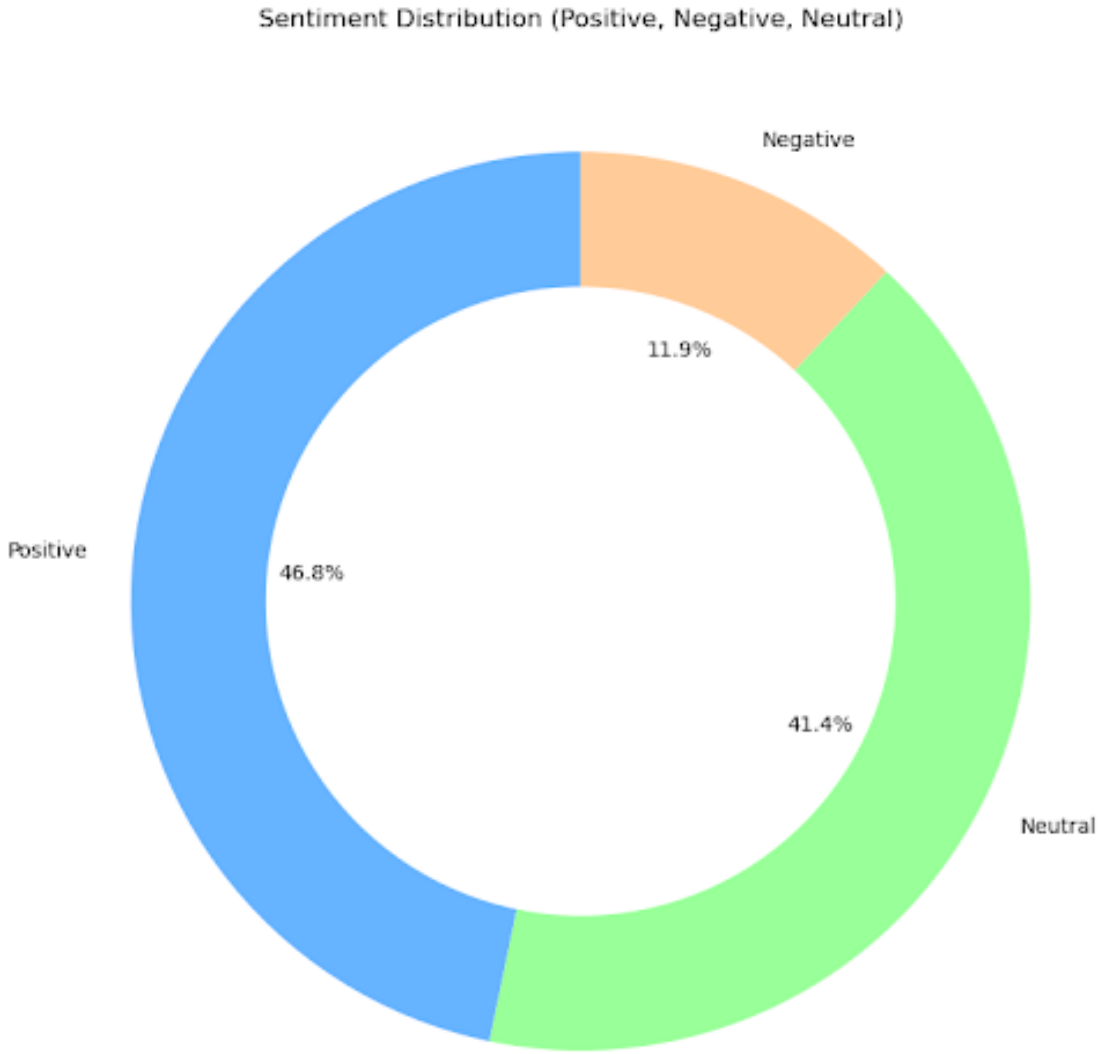


Fig. 4.1 Sentiment count

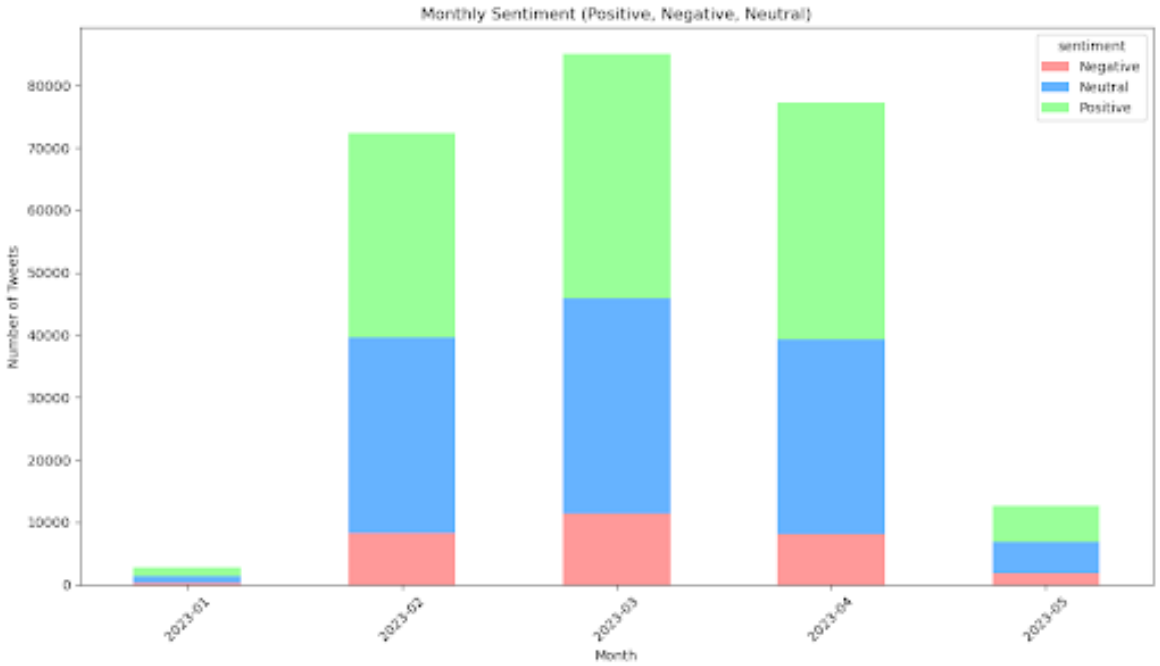


Fig. 4.2 Monthly Sentiment count

Above mentioned visualizations show the percentage of tweets for three different categories and also in another visualization, it shows monthly count of tweets as per its category.

4.3.2 Sentiment score vs Price of Bitcoin

This visualization signifies that Price of bitcoin is directly proportional to the sentiment of that particular day. The visualization consists of two line charts showing the relationship between BTC price and sentiment score over time. The top chart plots sentiment scores derived from NLP algorithms (TextBlob and VADER). Both sentiment metrics fluctuate above and below zero, representing positive and negative sentiments. It is clearly seen sharper spikes and dips for VADER. It shows that VADER is more sensitive towards textual tone.

While the second chart signifies the Adjusted close price for Bitcoin during the same period. It is clearly observable that change in the sentiment score is also noticeable in the Bitcoin price action. Together, these charts suggest a direct relationship between daily sentiment and price. Positive sentiment correlates with price increase, while negative aligns with decrease.

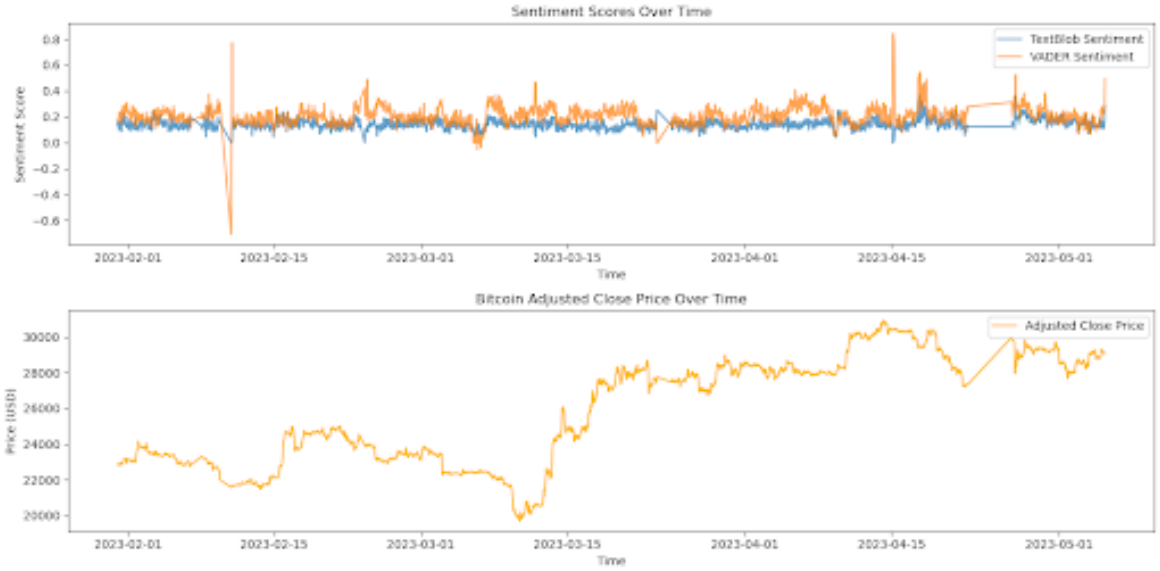


Fig. 4.3 Corelation between sentiments vs price

4.4 Model Development

4.4.1 Random Forest Classifier

Random forest provides a strong baseline model. It is a good starting point because it is a balance between simplicity, performance and flexibility. After transforming datasets, I was left with only quantitative data. My primary focus was on Bitcoin price, Where I introduced a new feature called the price indicator. This feature represents the difference between close and open prices of bitcoin. To simplify the prediction task, I converted the price indicator into a discrete variable: if the difference was negative, it was labeled as 0 and if positive, it was labeled as 1. It was reformulated into a binary classification task.

Class	**Precision**	**Recall**	**F1-Score**	**Support**
0	0.66	0.66	0.66	160
1	0.69	0.69	0.69	172
Accuracy		0.67 (332 samples)		
Macro Avg	0.67	0.67	0.67	332
Weighted Avg	0.67	0.67	0.67	332

Table 4.1 Classification Report - training

With hyperparameter tuning, There is a slight improvement in accuracy after tuning the hyperparameter.

Class	Precision	Recall	F1-Score	Support
0	0.61	0.77	0.68	217
1	0.65	0.47	0.55	198
Accuracy		0.63 (415 samples)		
Macro Avg	0.63	0.62	0.61	415
Weighted Avg	0.63	0.63	0.62	415

Table 4.2 Classification Report - testing

```
Iteration 1: n_estimators=200, max_depth=None, Accuracy=0.6386
Iteration 2: n_estimators=210, max_depth=None, Accuracy=0.6386
Iteration 3: n_estimators=220, max_depth=None, Accuracy=0.6386
Iteration 4: n_estimators=210, max_depth=None, Accuracy=0.6386
Iteration 5: n_estimators=220, max_depth=None, Accuracy=0.6386
Iteration 6: n_estimators=210, max_depth=None, Accuracy=0.6386
Iteration 7: n_estimators=200, max_depth=None, Accuracy=0.6386
Iteration 8: n_estimators=210, max_depth=None, Accuracy=0.6386
Iteration 9: n_estimators=200, max_depth=None, Accuracy=0.6386
Iteration 10: n_estimators=210, max_depth=None, Accuracy=0.6386
Iteration 11: n_estimators=220, max_depth=None, Accuracy=0.6386
Iteration 12: n_estimators=230, max_depth=None, Accuracy=0.6386
Iteration 13: n_estimators=240, max_depth=None, Accuracy=0.6386
Iteration 14: n_estimators=250, max_depth=None, Accuracy=0.6386
Iteration 15: n_estimators=240, max_depth=None, Accuracy=0.6386
Iteration 16: n_estimators=230, max_depth=None, Accuracy=0.6386
Iteration 17: n_estimators=220, max_depth=None, Accuracy=0.6386
Iteration 18: n_estimators=210, max_depth=None, Accuracy=0.6386
Iteration 19: n_estimators=220, max_depth=None, Accuracy=0.6386
Iteration 20: n_estimators=210, max_depth=None, Accuracy=0.6386

Best Parameters: n_estimators=210, max_depth=None, Best Accuracy=0.6386
```

Fig. 4.4 Random forest hyperparameter result

4.4.2 Random forest regressor

Since It is a classification problem and target variables should be continuous values then only Regressor could work for this model. I have also tried SVR (Support vector regressor) for

this dataset, it did not work at all and shows negative R2. It means the model is not able to predict any relations between target variable and predicted value.

```
y_pred_reg = rf_regressor.predict(x_val)

# Evaluate regression metrics
mae = mean_absolute_error(y_val, y_pred_reg)
mse = mean_squared_error(y_val, y_pred_reg)
r2 = r2_score(y_val, y_pred_reg)

mae, mse, r2
```

```
(0.4085090361445783, 0.19226332831325302, 0.22994065770348837)
```

Fig. 4.5 Random forest regressor

4.5 Model Development with Lag feature

To enhance the study, Introduced new lagged features and technical financial indicators into the dataset. The process began with loading the VADER sentiment analysis data along with Bitcoin hourly price data. These datasets were then merged to align BTC price with sentiment score. For each hour, the average sentiment score was calculated, along with the total number of tweets containing positive and negative sentiments.

btc_data_merged

	Datetime	Adj Close	Close	High	Low	Open	Volume	avg_sentiment_score	sentiment_positive	sentiment_negative	total_tweets
0	2023-01-31 00:00:00	22857.76563	22857.76563	22864.25195	22765.56836	22840.79688	0	0.165558	7.0	0.0	12.0
1	2023-01-31 01:00:00	22858.32813	22858.32813	22870.33789	22797.08398	22855.97852	0	0.178608	49.0	15.0	119.0
2	2023-01-31 02:00:00	22893.31836	22893.31836	22909.50195	22847.17969	22855.14844	0	0.186917	54.0	16.0	120.0

Fig. 4.6 merged sentiment added features - average sentiment score, positive tweets, negative tweets

Additionally, lagged sentiment features were created, capturing sentiment values lagged by 1, 2, 3 hours. This approach aims to explore potential relationships between past sentiment trends and BTC price movements

#	Column	Non-Null Count	Dtype
0	Datetime	2226 non-null	datetime64[ns]
1	Adj Close	2226 non-null	float64
2	Close	2226 non-null	float64
3	High	2226 non-null	float64
4	Low	2226 non-null	float64
5	Open	2226 non-null	float64
6	Volume	2226 non-null	int64
7	avg_sentiment_score	2072 non-null	float64
8	sentiment_positive	2072 non-null	float64
9	sentiment_negative	2072 non-null	float64
10	total_tweets	2072 non-null	float64
11	avg_sentiment_score_lag1	2072 non-null	float64
12	sentiment_positive_lag1	2072 non-null	float64
13	sentiment_negative_lag1	2072 non-null	float64
14	avg_sentiment_score_lag2	2072 non-null	float64
15	sentiment_positive_lag2	2072 non-null	float64
16	sentiment_negative_lag2	2072 non-null	float64
17	avg_sentiment_score_lag3	2072 non-null	float64
18	sentiment_positive_lag3	2072 non-null	float64
19	sentiment_negative_lag3	2072 non-null	float64

Fig. 4.7 Lagged features info

4.5.1 Random Forest Regressor with Sentiment Lag features

As per the output still the model is not getting near to an acceptable range of prediction. Still the Mean Absolute Error (MAE) is 2146.42 USD, that means actual value is off by 2146 USD from the predicted value.

on the other hand Root Mean Squared Error is 2506, that means on average error between actual and predicted value is high.

```
from sklearn.model_selection import GridSearchCV
from sklearn.ensemble import RandomForestRegressor
model_val = RandomForestRegressor(n_estimators=100, random_state=42)
model_val.fit(X_train, y_train)
y_pred_val = model_val.predict(x_val)
mae_val = mean_absolute_error(y_val, y_pred_val)
rmse_val = np.sqrt(mean_squared_error(y_val, y_pred_val))

mae_val, rmse_val
```

```
(2146.420749650305, 2506.717535178123)
```

Fig. 4.8 Random forest regressor validate - lagged features

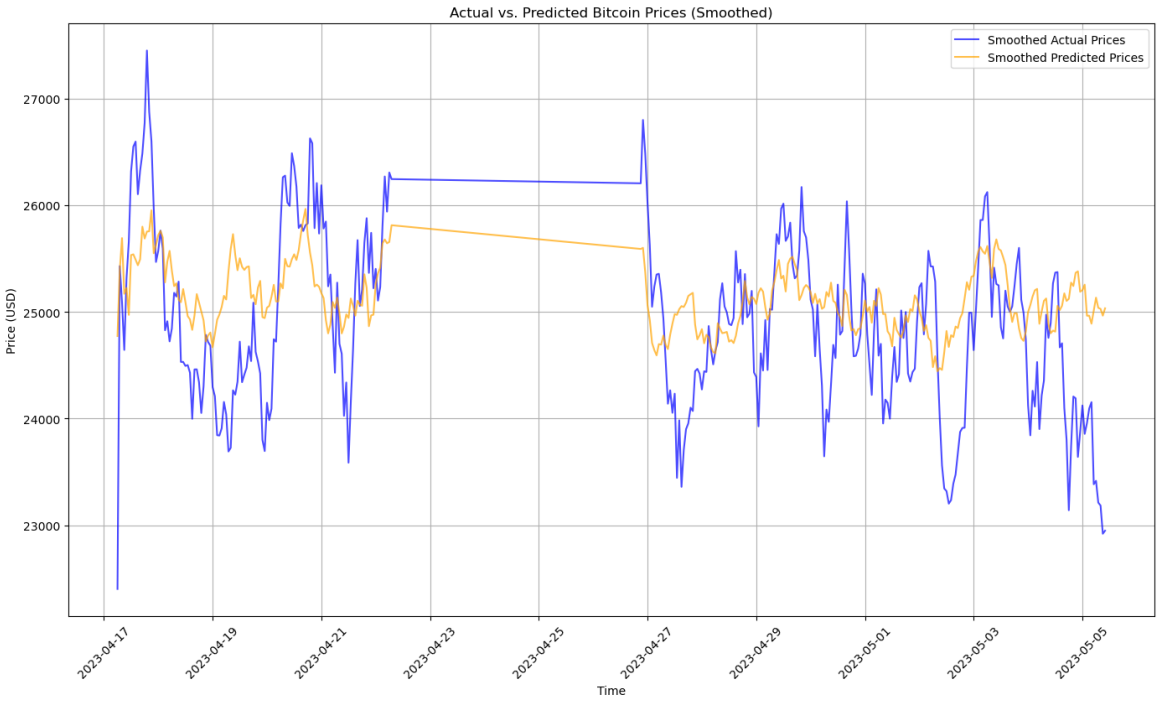


Fig. 4.9 Actual vs Predicted (Smooth)

With hyperparameter tuning there was no effect on the result, just a slight few USD difference achieved.

4.5.2 Random Forest Regressor with Sentiment Lag features + Technical Financial Indicators (Simple Moving Average, RSI)

To improve the prediction power of the model, technical indicators were introduced into the dataset. These indicators were calculated on the basis of Adj close price.

SMA (Simple moving average)

- 10 hour rolling window of Adj close price to capture short term price
- 30 hour SMA for capturing longer term price movements

Volatility

- Standard deviation of the Adjusted close price over 10 hour window was calculated to measure price fluctuations

RSI (Relative Strength Index)

- It indicates the overbought or oversold condition in the market

The new features were added into the model SMA_10, SMA_30, volatility, RSI, In addition the sentiment lag features.

```
# Evaluate the model
mae_optimized_indicators_opt = mean_absolute_error(y_test, y_pred_optimized_indicators_opt)
rmse_optimized_indicators_opt = np.sqrt(mean_squared_error(y_test, y_pred_optimized_indicators_opt))
r2__optimized_indicators_opt = r2_score(y_test, y_pred_optimized_indicators_opt)

mae_optimized_indicators_opt, rmse_optimized_indicators_opt, r2__optimized_indicators_opt
```

```
(379.12356502341294, 493.624505415528, 0.6497349965892893)
```

Fig. 4.10 regressor technical indicator result

This model performs significantly better than earlier models. The reduced MAE and RMSE, along with R2 0.65, demonstrate that the additional features contributed to increased predictive accuracy. SMA, RSI actually enhanced the model performance to capture trends.

4.6 Conclusion

Random Forest Classifier It has produced satisfactory outcomes for binary classification. The model exhibited an accuracy of 0.67 on validation datasets and 0.63 on testing datasets.

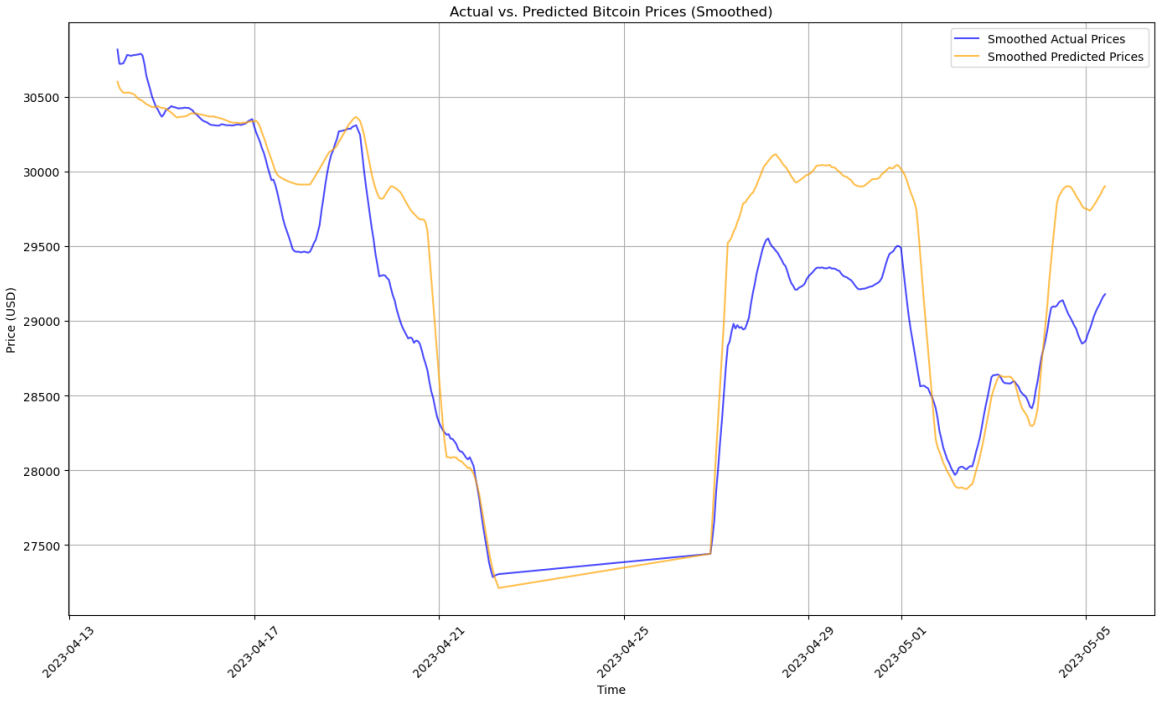


Fig. 4.11 regressor technical indicator result - Actual vs predicted

The classification report revealed that although precision and recall were fairly even, the model encountered some difficulties in predicting positive price as recall was lower.

Regression Approach

Random forest regressor and support vector regressor, did not perform effectively. Negative R square signifying poor model performance. Given that the target variable is categorical, the regression approach was not suitable. A classification model is more suitable in this scenario.

On the other hand lagged features, particularly sentiment data from VADER analysis showed some potential for enhancing the model. It provided that there was a relation between BTC price and Sentiments. Although the model improved with these lagged features , the performance was still below acceptable level because Mean Absolute Error and Root mean squared error remain high which is not an ideal case for a regression model.

When additional technical indicators such as Simple Moving Average (SMA), Relative Strength Index (RSI), and volatility (Standard deviation) were added, the model performance improved significantly. The addition of financial indicators strengthened the model's predictive ability

At the end of Chapter 4, I would like to say that Random Forest Regressor with technical indicators and lagged sentiment features provided the best result. The enhanced model

performance, reflected in decreased MAE and RMSE, indicated that merging sentiment analysis with technical indicators help formulate a more resilient model for predicting Bitcoin price.

Chapter 5

Conclusion and Future Direction

In the final chapter, I concluded analysis of sentiment in Bitcoin tweets data. By combining sentiment analysis with technical indicators, the study sought to improve the predictability of Bitcoin price fluctuations.The study involved the application of TextBlob and VADER sentiment analysis tools, data preprocessing, feature engineering and implementation of machine learning models (Random forest classifier and Random forest regressor)

5.1 Summary of findings

5.1.1 Model Performances

This research provided the impact of public opinion derived from tweets on bitcoin price trends utilizing machine learning techniques. Sentiment evaluation through TextBlob and VADER indicates that sentiment analysis correlated with the price of the Bitcoin. Feature development, encompassing lagged sentiment metrics and technical indicators such as SMA, RSI, and variability for adjusted close price. Classification algorithms, especially the Random Forest classifier, displayed moderate performance with 67%, where the Regression model faced some difficulties. Merging sentiment analysis and technical indicators together illustrated potential for more reliable Bitcoin price forecasts.

5.1.2 Visual Representation

Visualization such as sentiment counts, price and sentiment relation, actual vs predicted price

- Sentiment Distribution (Doughnut chart illustrate the proportion of positive, neutral and negative tweets sentiment in the dataset)

- Monthly sentiment trends (Bar chart) - Monthly sentiments counts are shown, categorizing tweets by their sentiment types
- Sentiment score vs Bitcoin price (Line chart) - Two line chart display fluctuation of sentiment score and bitcoin price over the same period of time.
- Actual vs predicted price

5.2 Implications and Recommendations

The findings of this study provides significant consequences for both academic and practical use in digital currency markets. By utilizing sentiment analysis and machine learning algorithms, this study emphasizes the complex connection between public sentiment and Bitcoin price variation. The incorporation of sentiment metrics with technical financial indicators, such as moving averages and RSI, provides a unique strategy for improving predictive precision.

5.2.1 Implications

The demonstrated correlation between sentiment scores and Bitcoin price suggest that traders can integrate sentiment analysis into their decision making process for getting better market trends. On the hand of machine learning practitioners, this study provides the importance of incorporating lagged sentiment features and financial indicators to improve predictive modeling performance.

The finding highlights the limitation of regression based approaches in scenarios where target variables exhibit categorical behaviour.

This study provides a framework for understanding how public sentiment captured from twitter influences market behavior. This understanding is critical for assessing market risks and developing strategies to mitigate the effect of price volatility.

5.2.2 Recommendation

To promote the reach and applicability of effective analysis in the cryptocurrency market, several recommendations can be proposed. Improving data acquisition and analysis is paramount. Future research should focus on enlarging the dataset by integration text data from various social media platforms, new articles and discussion forums. This enhancement would assist in developing a more holistic sentiment index, broader array of public opinion.

Enhancing model precision is an important domain for advancement, Utilizing profound learning strategies. such as Long short term memory or Gated recurrent unit, can effectively capture temporal dynamics. In addition Investring adaptive feature development techniques such as sentiment analysis on the basis of popularity of tweets or influencer effect, could improve the reactiveness of predictive model.

5.3 Limitation

5.3.1 Data Limitations

The research primarily utilized tweets from twitter which, although important, constitutes merely one facet of public sentiment. There are many other platforms like Reddit, News, Forums and Telegram groups that may have better settlement indexes.

5.3.2 Data Quality

The presence of noisy, unwanted and spam content within the dataset presented difficulties for the sentiment analysis.

5.4 Challenges

5.4.1 Bitcoin Volatile Nature

Bitcoin extreme price volatility, influenced by various external factors such as regulatory authorities decisions and economic trends, created an unpredictability that sentiment analysis alone is unable to fully capture. A single news can change the market sentiment from bull to bear and vice versa.

5.4.2 Model Performance

While Random Forest Classifier and regressors provided moderate success, the regression models struggled to achieve acceptable accuracy with high error metrics. After relying on lagged features and technical indicators showed promise for regression but still insufficient to fully address the complexity of price fluctuation.

5.4.3 Temporal Dynamics

Capturing the dynamic nature of sentiment and its delayed impact on prices proved challenging. Sentiment may not immediately translate into price change, leading to potential mismatches in the data. In Spite of above mentioned challenges

In addition to the challenges outlined, several other features such as hash rates, transaction fees, network activity, and transaction counts also affect Bitcoin's price. Implementing these features into predictive models would provide a more holistic understanding of price prediction. The integration of these features represent an opportunity to understand the Bitcoin price movement in a deeper way.

5.4.4 Practical challenges

This study actually shows potential for real time sentiment analysis, implementing and maintaining such a system would require high cost and best infrastructure. This is the only reason I was not able to perform Long short term memory (LSTM) model for my research. Because my laptop was not able to run the tensorflow library because of some DLL file issue. I tried to run this on Google Colab, and found it was taking so long to execute the machine learning model.

5.5 Conclusion

This research explores the integration of sentiment analysis and machine learning for predicting Bitcoin price fluctuations, highlighting the importance of social media views in market dynamics. Utilizing tools such as TextBlob and VADER, along with technical financial metrics like simple moving average, RSI and volatility, the research illustrates the capability of hybrid models to improve predictive accuracy. Although Random Forest Classifiers show moderate effectiveness, the inclusion of lagged sentiments and financial indicators significantly enhanced results. Issues such as data integrity, bitcoin volatile nature and computational constraint highlight the complexity of this domain. Finally, I say future research should focus on large datasets and advanced modeling techniques.

References

[1] I. Georgoula, D. Pournarakis, C. Bilanakos, D. N. Sotiropoulos, and G. M. Giaglis, "Using Time-Series and Sentiment Analysis to Detect the Determinants of Bitcoin Prices," *MCIS 2015 Proceedings*, 2015, Paper 20. Available at: https://aisel.aisnet.org/mcis2015/20.

[2] D. R. Pant, P. Neupane, A. Poudel, A. K. Pokhrel, and B. K. Lama, "Recurrent Neural Network Based Bitcoin Price Prediction by Twitter Sentiment Analysis," *2018 IEEE 3rd International Conference on Computing, Communication and Security (ICCCS)*, Kathmandu, Nepal, 2018, pp. 128-132, doi: 10.1109/CCCS.2018.8586824.

[3] V. Pagolu, K. Challa, and G. Panda, "Sentiment Analysis of Twitter Data for Predicting Stock Market Movements," *International Conference on Signal Processing Communication Power and Embedded System (SCOPES)*, 2016.

[4] H. Htun, M. Biehl, and N. Petkov, "Survey of feature selection and extraction techniques for stock market prediction," *Financial Innovation*, vol. 9, no. 1, Jan. 2023. Available at: https://doi.org/10.1186/s40854-022-00441-7.

[5] X. Cai, S. Hu, and X. Lin, "Feature extraction using restricted Boltzmann machine for stock price prediction," *IEEE CSAE*, Zhangjiajie, China, 2012, pp. 80–83. J. Chen, "Analysis of Bitcoin Price Prediction Using Machine Learning," Journal of Risk and Financial Management, vol. 16, no. 1, p. 51, Jan. 2023, https://www.mdpi.com/1911-8074/ 16/1/51.

[6] I. Gurrib and F. Kamalov, "Predicting Bitcoin price movements using sentiment analysis: A machine learning approach," *Studies in Economics and Finance*, Emerald Group Publishing Limited, vol. 39, no. 3, pp. 347–364, December 2021. Available at: https://ideas.repec.org/a/eme/sefpps/sef-07-2021-0293.html.

[7] R. Ramneet, D. Gupta, and M. Madhukar, "Title Unknown," *Journal of Computational and Theoretical Nanoscience*, vol. 17, no. 9-10, pp. 4535-4542, September/October 2020. American Scientific Publishers. Available at: https://doi.org/10.1166/jctn.2020.9300.

[8] G. Serafini *et al.*, "Sentiment-Driven Price Prediction of the Bitcoin based on Statistical and Deep Learning Approaches," *IEEE Xplore*, Jul. 01, 2020. Available at: https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=9206704.

[9] S. Singh, A. Pise, and B. Yoon, "Prediction of bitcoin stock price using feature subset optimization," *Heliyon*, vol. 10, no. 7, p. e28415, Apr. 2024. Available at: https://doi.org/10.1016/j.heliyon.2024.e28415.

Appendix A

Data Preparation, Preprocessing & Transformation

A.1 Import Libraries

The figure below shows the code screenshot to import the python libraries used to prepare and transform the data.

```
# Filtering out the warnings
import warnings

warnings.filterwarnings('ignore')
import pandas as pd  # Data manipulation
import numpy as np  #Numerical computation
import matplotlib.pyplot as plt # Plotting
import seaborn as sns # Statistical Data visualtization
import re # Regular expression
from vaderSentiment.vaderSentiment import SentimentIntensityAnalyzer # sentiment analysis
import yfinance as yf # Yahoo finance for BTC price extract
import ta # For adding technical Indicators SMA, RSI etc.
from sklearn.model_selection import train_test_split
from sklearn.metrics import classification_report, confusion_matrix, accuracy_score
from nltk.stem.porter import PorterStemmer
from nltk.stem import WordNetLemmatizer
from textblob import TextBlob
from sklearn.ensemble import RandomForestClassifier
from sklearn.model_selection import RandomizedSearchCV
from imblearn.over_sampling import SMOTE
from sklearn.svm import SVR
from sklearn.model_selection import GridSearchCV
from sklearn.ensemble import RandomForestRegressor
from sklearn.metrics import mean_absolute_error, mean_squared_error, r2_score
```

Fig. A.1 Python libraries

A.2 Reading Dataset

After Importing libraries, need to read the dataset using Pandas library

```
tweets = pd.read_csv('tweets.csv')
```

```
tweets.head()
```

A.3 Data transformation

First thing have done changing the timestamp to nearest hour to make it compatible to bitcoin hourly data.

Transform the data using apostrophe_dict and emotion_dict to change the words i.e. you'll' to you will and change the emoticon to actual words.

`# Apostrophe Dictionary`
`apostrophe_dict = {`
`"ain't": "am not / are not",`
`"aren't": "are not / am not",`
`"can't": "cannot",`
`"can't've": "cannot have", ...`
`def contx_to_exp(text):`
`  for key in apostrophe_dict:`
`  value = apostrophe_dict[key]`
`  text = text.replace(key,`
`  value)`
`  return text`
	`":(": "sad",`
	`":-c": "sad",`
	`":-< ": "sad",`
	`":-[": "sad",`
	`":-||": "sad"`
	`}`
	
	`def emotion_check(text):`
	`  for key in emotion_dict:`
	`  value = emotion_dict[key]`
	`  text = text.replace(key,`
	`  value)`
	`  return text`

Table A.1 Text transformation

```
def clean_text(text):
  text = re.sub(r'https?:\/\/\S*'," ", text) # Removing the url from the text
  text = re.sub(r'@\S+', " ", text) # Removing twitter handles from the text
  text = re.sub('#'," ", text) # removing # from the data
  text = re.sub(r'RT', "", text) # Removing the Re-tweet mark
  text = re.sub(r"\s+"," ", text)  # Removing Extra Spaces
  text = text.lower()
  return text

import re
def remove_pattern(input_txt, pattern):
    r = re.findall(pattern, input_txt)
    for word in r:
        input_txt = re.sub(word, "", input_txt)
    return input_txt.lower()
```

Fig. A.2 text cleaning

After that new column name Clean_tweet created in the dataset

```
tweets['clean_tweet'] = np.vectorize(remove_pattern)(tweets['text'], "@[\w]*")

#using above functions
tweets['clean_tweet'] = tweets['clean_tweet'].apply(lambda x : clean_text(x))
tweets['clean_tweet'] = tweets['clean_tweet'].apply(lambda x : contx_to_exp(x))
tweets['clean_tweet'] = tweets['clean_tweet'].apply(lambda x : emotion_check(x))

#removing special characters, numbers and punctuations
tweets['clean_tweet'] = tweets['clean_tweet'].str.replace("[^a-zA-Z]", " ")


#remove short words
tweets['clean_tweet'] = tweets['clean_tweet'].apply(lambda x: " ".join([w for w in x.split() if len(w)>3]))

# Removing every thing other than text
tweets['clean_tweet'] = tweets['clean_tweet'].apply( lambda x: re.sub(r'[^\w\s]',' ',x))  # Replacing Punctuations with space
tweets['clean_tweet'] = tweets['clean_tweet'].apply( lambda x: re.sub(r'[^a-zA-Z]', ' ', x)) # Replacing all the things with space other than text
tweets['clean_tweet'] = tweets['clean_tweet'].apply( lambda x: re.sub(r"\s+"," ", x)) # Removing extra spaces


#individual words as tokens
tokenized_tweet = tweets['clean_tweet'].apply(lambda x: x.split())


#stem the words

from nltk.stem.porter import PorterStemmer
from nltk.stem import WordNetLemmatizer
stemmer = PorterStemmer()
lemmatizer = WordNetLemmatizer()

tokenized_tweet = tokenized_tweet.apply(lambda sentence: [lemmatizer.lemmatize(stemmer.stem(word)) for word in sentence])



#combine words into single sentence
for i in range(len(tokenized_tweet)):
    tokenized_tweet[i] = " ".join(tokenized_tweet[i])



tweets['clean_tweet'] = tokenized_tweet
tweets.head()
```

A.4 Sentiment Analysis and Visualizations

After cleaning tweets, and removing all irrelevant signs, words, we are moving to the next step, which is sentiment analysis. First we will be looking snapshot for TextBlob and after that VADER

```
from textblob import TextBlob     # for performing NLP Functions
polarity=[]     #list that contains polarity of tweets
subjectivity=[]    ##list that contains subjectivity of tweets

for i in tweets.text.values:
    try:
        analysis = TextBlob(i) # [i] records to the first data in dataset
        polarity.append(analysis.sentiment.polarity)
        subjectivity.append(analysis.sentiment.subjectivity)

    except:
        polarity.append(0)
```

```
        subjectivity.append(0)



# adding sentiment score in csv

tweets['sentiment_score'] = polarity
tweets.head()
```

Now we have implemented code for VADER

```
# Apply sentiment analysis to the 'clean_tweet' column
# Convert non-string values to NaN, then fill with an empty string or drop
tweets['clean_tweet'] = tweets['clean_tweet'].fillna('').astype(str)

# Apply the sentiment analysis function
tweets['sentiment_score'] = tweets['clean_tweet'].apply(get_vader_sentiment)

# Classify the sentiment as positive, neutral, or negative
def classify_vader_sentiment(score):
    if score > 0.05:
        return 'Positive'
    elif score < -0.05:
        return 'Negative'
    else:
        return 'Neutral'
```

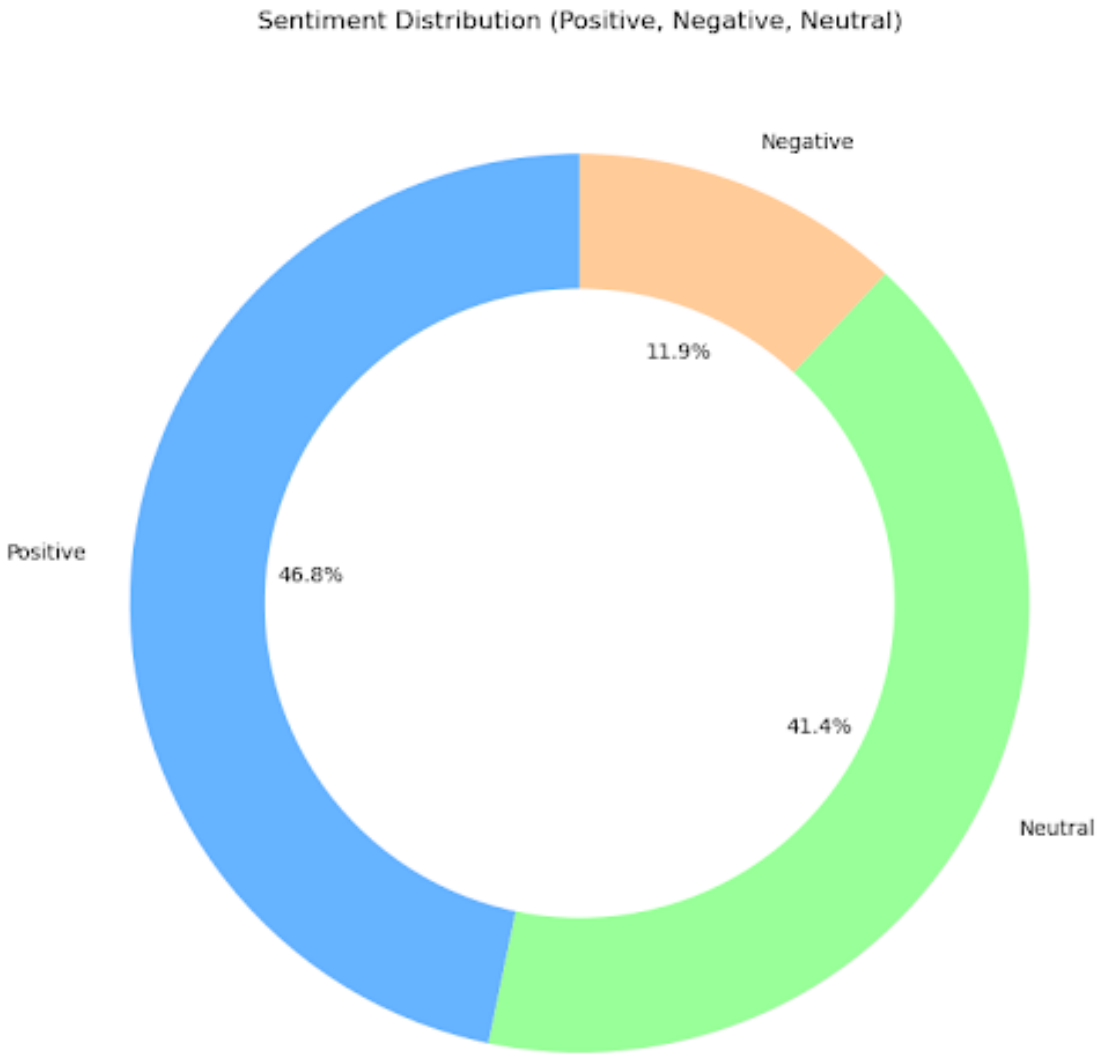
```
import matplotlib.pyplot as plt

# Load the dataset
#textblob_df = pd.read_csv('tweets_with_vader_sentiment.csv')

# Count the sentiment categories
sentiment_counts = tweets['sentiment'].value_counts()

# Create a doughnut plot
plt.figure(figsize=(8, 8))
plt.pie(sentiment_counts, labels=sentiment_counts.index, autopct='%1.1f%%', startangle=90, colors=['#66b3ff', '#99ff99', '#ffcc99'])
center_circle = plt.Circle((0,0), 0.70, fc='white')  # Create a doughnut effect
plt.gca().add_artist(center_circle)

# Add a title
plt.title('Sentiment Distribution (Positive, Negative, Neutral)')
plt.tight_layout()
plt.show()
```



A.5 Bitcoin Price Data

BTC - USD hourly basis data extracted during the specific time period.

After matching timestamp format with VADER and TextBlob Sentiment, we merged the data. Before merging I performed a few tricks and tweaks to change the date time format the same for all datasets.

After merging, a new feature announced name Price_ indicator that signify the price difference in close price - open price, on the basis of that classification model implemented.

[18]:
```
# Load the TextBlob and VADER sentiment datasets
textblob_df = pd.read_csv('Cleaned tweets with sentiments.csv')
vader_df = pd.read_csv('tweets_with_vader_sentiment.csv')
def parse_date(date_str):
    try:
        return pd.to_datetime(date_str, format='%m/%d/%Y-%H:%M')
    except ValueError:
        return pd.to_datetime(date_str, format='%m/%d/%Y-%H:%M')

textblob_df['rounded_time'] = textblob_df['rounded_time'].apply(parse_date)
vader_df['rounded_time'] = vader_df['rounded_time'].apply(parse_date)

# Ensure the 'date' column is in datetime format for proper plotting, with a specified format
textblob_df['rounded_time'] = pd.to_datetime(textblob_df['rounded_time'], format='%m/%d/%Y-%H:%M')
vader_df['rounded_time'] = pd.to_datetime(vader_df['rounded_time'], format='%m/%d/%Y-%H:%M')

# Aggregate sentiment scores by date (average sentiment per day)
textblob_daily = textblob_df.groupby('rounded_time')['sentiment_score'].mean().reset_index()
vader_daily = vader_df.groupby('rounded_time')['sentiment_score'].mean().reset_index()

# Ensure the 'Date' column in the Bitcoin dataset is in datetime format
df['rounded_time'] = pd.to_datetime(df['rounded_time'], format='%m/%d/%Y-%H:%M')

# Merge TextBlob, VADER, and Bitcoin data on 'date'
merged_data = pd.merge(textblob_daily, vader_daily, on='rounded_time', how='inner', suffixes=('_textblob', '_vader'))
merged_data = pd.merge(merged_data, df, left_on='rounded_time', right_on='rounded_time', how='inner')


# Show the first few rows of merged data
print(merged_data.head())
```

```
         rounded_time  sentiment_score_textblob  sentiment_score_vader  \
0 2023-01-31 00:00:00                  0.142336               0.165558
1 2023-01-31 01:00:00                  0.129171               0.178608
2 2023-01-31 02:00:00                  0.164494               0.186917
3 2023-01-31 03:00:00                  0.094387               0.158187
4 2023-01-31 04:00:00                  0.106855               0.235430

     Adj Close        Close         High          Low         Open  Volume
0  22857.76563  22857.76563  22864.25195  22765.56836  22840.79688       0
1  22858.32813  22858.32813  22870.33789  22797.08398  22855.97852       0
2  22893.31836  22893.31836  22909.50195  22847.17969  22855.14844       0
```

```
price_indicator = [merged_data.Close[0] - merged_data['Open'][0]]
for i in range(len(merged_data)-1):
    price_indicator.append(merged_data.Close[i+1] - merged_data.Close[i])
price_indicator
```

```
[16.96875,
 0.5625,
 34.99022999999943,
 -31.513670000000275,
```

Appendix B

Modeling and Results

B.1 Random forest classifier (Testing)

```
# Random Forest Classifier
baseline_model = RandomForestClassifier(random_state=42)
baseline_model.fit(X_train, y_train)

# Predictions and Evaluation
y_pred_classifier_test = baseline_model.predict(X_test)

# Metrics
accuracy = accuracy_score(y_test, y_pred_classifier_test)
report = classification_report(y_test, y_pred_classifier_test)


# Displaying the metrics in a readable format
print(f"Accuracy: {accuracy:.2f}")
print("\nClassification Report:\n")
print(report)
```

```
Accuracy: 0.63

Classification Report:

              precision    recall  f1-score   support

           0       0.61      0.77      0.68       217
           1       0.65      0.47      0.55       198

    accuracy                           0.63       415
   macro avg       0.63      0.62      0.61       415
weighted avg       0.63      0.63      0.62       415
```

After classification tasks, I have changed the target variable to continuous variable name adj close price and also introduced new features and lagged features such as sentiment lag, simple moving average, RSI and standard deviation.

B.2 Random forest Regressor with lagged and technical indicators (Testing)

```
# Fit the model with the best parameters
optimized_model_with_indicators_opt = RandomForestRegressor(
    max_depth=20,
    max_features='sqrt',
    min_samples_leaf=1,
    min_samples_split=2,
    n_estimators=200,
    random_state=42
)

# Train the model
optimized_model_with_indicators_opt.fit(X_train, y_train)

# Predict on the test set
y_pred_optimized_indicators_opt = optimized_model_with_indicators_opt.predict(X_test)

# Evaluate the model
mae_optimized_indicators_opt = mean_absolute_error(y_test, y_pred_optimized_indicators_opt)
rmse_optimized_indicators_opt = np.sqrt(mean_squared_error(y_test, y_pred_optimized_indicators_opt))
r2__optimized_indicators_opt = r2_score(y_test, y_pred_optimized_indicators_opt)

mae_optimized_indicators_opt, rmse_optimized_indicators_opt, r2__optimized_indicators_opt
```

```
(379.12356502341294, 493.624505415528, 0.6497349965892893)
```