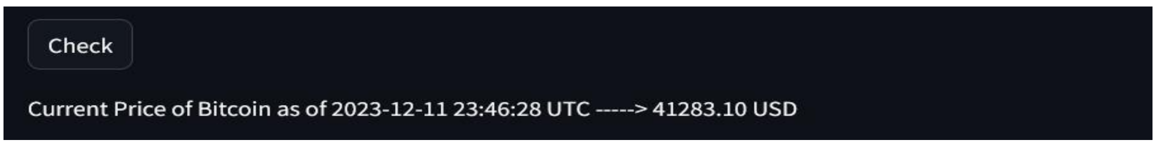
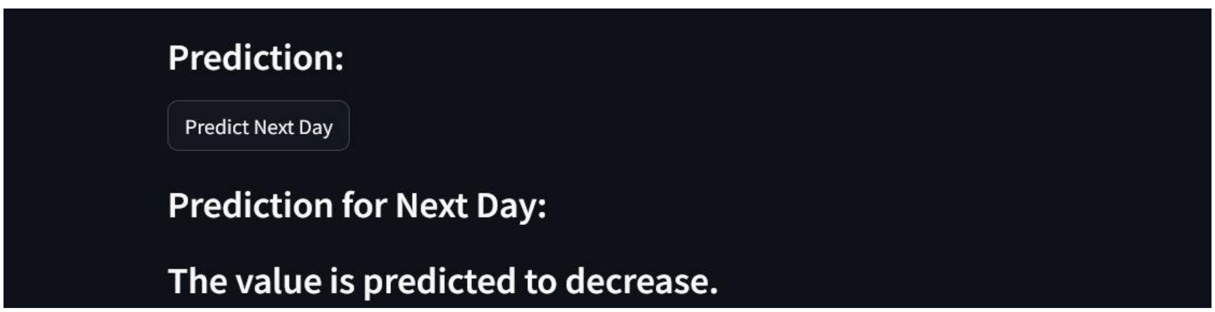
Cryptocurrency trend Predictor

Submitted by: Dev Karan Suresh, Sivani Dronamraju, Swathi Baskaran
DATA 602 – Principles of Data Science, Fall 2023
Professors: Mohammad T. Hajiaghayi
TA: Max Springer
December 12, 2023

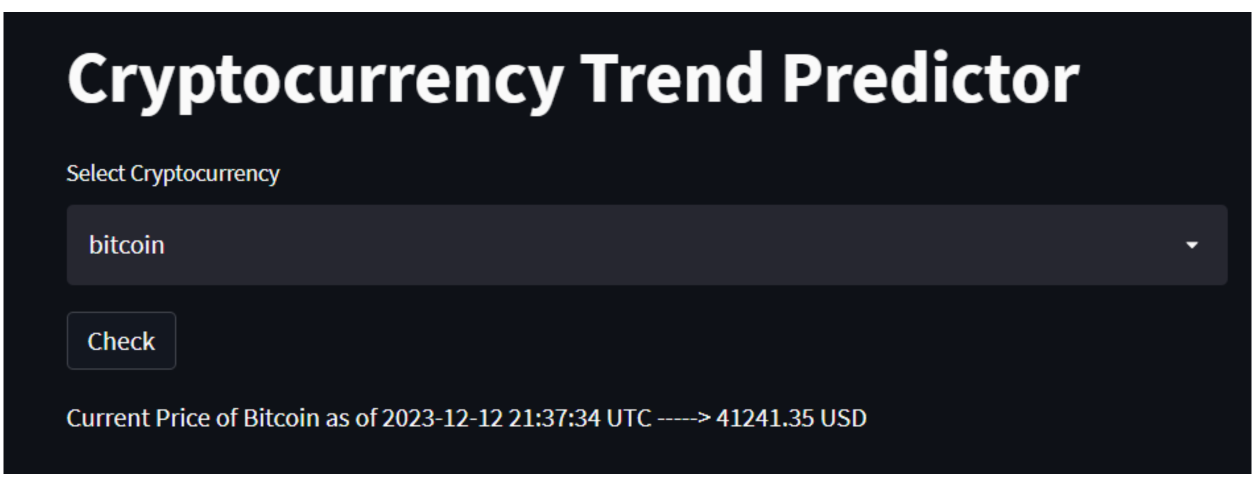
MODEL PREDICTION:



The Bitcoin Price was $41283.10 on 2023-12-11 at 23:46 UTC.



Our model predicted that the price is going to decrease on 2023-12-12.



The Price of Bitcoin is $41241.35 on 2023-12-12 at 21:37 UTC. It decreased.

i. Abstract

Our "Crypto Market Prediction" project employs innovative approaches to forecast cryptocurrency price trends, leveraging the PyCoinGecko API for real-time data updates every 1 - 5 minutes. By converting timestamps, aggregating daily data, and visualizing trends through candlestick graphs, the foundation is laid for machine learning models. By incorporating sentiment analysis, the project adapts to the evolving technological landscape by utilizing alternative.me's sentiment scores. Normalizing these scores and integrating them into the predictive model enhances accuracy. Logistic regression emerges as a reliable model, outperforming the other models in predicting general trends. While focusing on practicality, our project extends its analysis to incorporate the newsapi.org API, tapping into relevant news data to capture external influences on cryptocurrency markets.

Links to the Project:

WebApp: https://cryptocurrencytrendpredictor-fh6tvqeiuknhipp8edipke.streamlit.app/

Github: https://github.com/11swathi/Cryptocurrency_Trend_Predictor/tree/main

The deployed application has a very limited number of API calls, so it is very difficult to explore the deployed website without it throwing “rate limited errors”. One choice we had is to become a premium member of the CoinGecko API, which is reputable but it is pricey. Thus, we recommend you to use the GitHub repository to run the provided code directly.

ii. Introduction

Cryptocurrency, which uses block-chain technology to operate on decentralized networks, has completely changed the global financial scene. With Bitcoin being the most well-known, it provides transactions with immutability, security, and transparency. The market has drawn many enthusiasts due to its volatile nature and quick evolution. On the other hand, there are some risks associated with the market due to its characteristics. In this context, predicting market movements becomes more important for informed decision-making. Our project seeks to provide insights into cryptocurrency price trends by leveraging a multifaceted approach. Because of the market’s speculating character and sensitivity to a wide range of circumstances, cryptocurrency price patterns provide a complex maze that requires advanced analytical tools to explore. Because working with time series may be so complicated, it is important to take the continuous nature of the data into account (Felizardo et al., 2019). So, we chose to work with fewer coins in order to increase the reliability of our project. To improve the precision and scope of market forecasts, we combined sentiment analysis, real-time data, and news sources in this project.

Additionally, the use of candlestick graphs as a visualization method makes it easy for the users with minimum knowledge to understand the market trend better. The integration and normalization of sentiment scores emphasize the critical role that machine learning and sentiment analysis play in forecasting price changes. This can be used to find out what other people's views are about the cryptocurrency market (Wimalagunaratne & Poravi, 2018). Additionally, a comparative analysis of several machine learning techniques in our project made it clear that logistic regression is a dependable modeling methodology, recognizing the difficulties in making accurate forecasts in these wild markets. In our project we acknowledged the complex interactions between global economic concerns, changing regulations, and technology breakthroughs that influence the cryptocurrency environment.

1. Model Methodologies

1.1.1 Data Source

The PyCoinGecko API, a potent interface for real-time cryptocurrency data, is the project's mainstay. We can dynamically download the most recent cryptocurrency price data thanks to this Python API, which enables smooth contact with the CoinGecko platform. We particularly obtain historical market chart data in USD for the last ninety days.

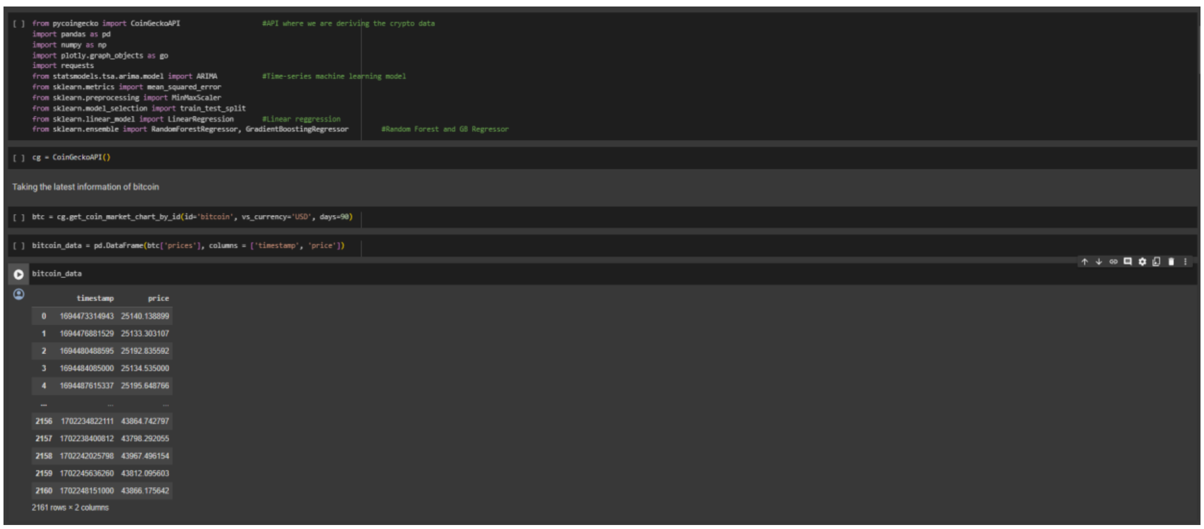


Figure i

1.1. 2 Timestamp Conversion and Daily Aggregation

Preprocessing is an essential step for improved interpretability of the Bitcoin data that is produced, which at first contains timestamps in a machine-readable format. These timestamps are transformed into a datetime format that can be read by humans using the pandas package. In order to provide time-based analysis and give a clear and intuitive grasp of the temporal aspect of market movements, this conversion is essential.

Converting timestamp to datetime format

```
bitcoin_data['Date'] = pd.to_datetime(bitcoin_data['timestamp'], unit = 'ms')
```

```
[ ] bitcoin_data
```

	timestamp	price	Date
0	1694473314943	25140.138899	2023-09-11 23:01:54.943
1	1694476881529	25133.303107	2023-09-12 00:01:21.529
2	1694480488595	25192.835592	2023-09-12 01:01:28.595
3	1694484085000	25134.535000	2023-09-12 02:01:25.000
4	1694487615337	25195.648766	2023-09-12 03:00:15.337
...	...	...	...
2156	1702234822111	43864.742797	2023-12-10 19:00:22.111
2157	1702238400812	43798.292055	2023-12-10 20:00:00.812
2158	1702242025798	43967.496154	2023-12-10 21:00:25.798
2159	1702245636260	43812.095603	2023-12-10 22:00:36.260
2160	1702248151000	43866.175642	2023-12-10 22:42:31.000

2161 rows × 3 columns

Figure ii

After the conversion of timestamps, the information is carefully combined. We create daily aggregates by grouping the data by date using the pandas package. The computation of the maximum, minimum, opening, and closing prices for each day is a necessary step in this aggregate. Condensing the raw data into a more digestible manner, the resulting dataset assumes an organized form. This combined dataset captures the everyday subtleties of cryptocurrency market behavior and acts as an anchor for further modeling and research.

Aggregating all the values on the basis of date and creating a new dataframe. We will use this for our prediction and visualization.

```
[ ] candlestick_data = bitcoin_data.groupby(bitcoin_data.Date.dt.date).aggregate({'price': {'max', 'min', 'first', 'last'}})
```

```
[ ] candlestick_data
```

	price			
	last	first	max	min
Date				
2023-09-11	25140.138899	25140.138899	25140.138899	25140.138899
2023-09-12	25979.215385	25133.303107	26288.334884	25133.303107
2023-09-13	26234.189828	25850.322300	26304.015465	25843.107805
2023-09-14	26615.532123	26222.013304	26751.796646	26193.031946
2023-09-15	26792.608825	26531.395566	26792.608825	26251.615685
...	...	...	...	...
2023-12-06	43634.904484	44105.944773	44123.756422	43613.258584
2023-12-07	43166.048404	43788.288808	43985.767023	43158.794425
2023-12-08	44252.835095	43270.119560	44385.975286	43174.140967
2023-12-09	43908.530003	44158.481190	44290.263279	43824.945022
2023-12-10	43866.175642	43714.935354	43968.397400	43634.411927

91 rows × 4 columns

Figure iii

1.1. 3 Visual Representation

Finding patterns in large, complicated datasets is made possible in a significant way by the use of visualization. We use the Plotly library in our project to create visually appealing candlestick charts from the aggregated data. These graphs provide a thorough overview of the daily changes in cryptocurrency prices during the previous ninety days. The opening, closing, maximum, and minimum prices are graphically represented in candlestick charts, which offer a comprehensive and perceptive representation of market patterns. This graphic depiction is a useful tool for communicating as well as for spotting trends. The complexities of the crypto market dynamics are easily understood by stakeholders, which promotes a deeper comprehension of the trends revealed by our analysis.

Figure iv

1.2 Sentiment Analysis

1.2.1 Initial Plan

The initial approach for the project involved utilizing Tweepy to leverage the Twitter API as it progressed to include sentiment analysis. Using this method, one can ascertain the opinions of others on a given subject.

Our goal was to collect and integrate real-time social media data, particularly tweets related to cryptocurrency movements because we recognized the priceless insights that can be found there. Due to the changes to Twitter's access regulations, we were unable to retrieve the required data. This made it necessary to assess potential backup plans in order to guarantee the sentiment analysis portion of our project.

1.2. 2 Alternative Solution

We were looking for alternative solutions since we couldn't retrieve data from Twitter API, and that is when we came across alternative.me API, which could help us with sentiment analysis. This API offered a workable solution by offering sentiment scores, which ranged from 0 to 100 and covered the emotional range of the cryptocurrency market. These scores enhance our comprehension of emotional dynamics by providing detailed insights into market attitudes ranging from fear to greed.

Taking the sentiment data from alternative.me

```
[ ] url = 'https://api.alternative.me/fng/?limit=91&date_format=cn'

    r= requests.get(url)
```

```
[ ] r
```

```
<Response [200]>
```

```
[ ] data = r.json()
```

```
[ ] temp_df = pd.DataFrame(data['data'])
```

```
[ ] temp_df
```

	value	value_classification	timestamp	time_until_update
0	74	Greed	2023-12-10	-1702160477
1	73	Greed	2023-12-09	NaN
2	72	Greed	2023-12-08	NaN
3	72	Greed	2023-12-07	NaN
4	72	Greed	2023-12-06	NaN
...	...	...	...	...
86	45	Fear	2023-09-15	NaN
87	45	Fear	2023-09-14	NaN
88	41	Fear	2023-09-13	NaN
89	30	Fear	2023-09-12	NaN
90	40	Fear	2023-09-11	NaN

91 rows × 4 columns

Figure v

To guarantee the durability of our predictive model, a two-step approach was carefully implemented to incorporate sentiment analysis.

First, normalization was used to put sentiment ratings into a defined range of -1 to 1, harmonizing them with other numerical features in our dataset. This alignment guarantees that our predictive model is compatible with machine learning techniques and makes the integration of sentiment analysis easier.

Rescaling sentiment from 0 - 100 to -1 to 1

```
[ ] temp_df['value'] = temp_df['value'].astype(int)
```

```
[ ] temp_df['timestamp'] = pd.to_datetime(temp_df['timestamp'])
```

```
[ ] temp_df
```

	value	value_classification	timestamp	time_until_update
0	74	Greed	2023-12-10	-1702160477
1	73	Greed	2023-12-09	NaN
2	72	Greed	2023-12-08	NaN
3	72	Greed	2023-12-07	NaN
4	72	Greed	2023-12-06	NaN
...	...	...	...	...
86	45	Fear	2023-09-15	NaN
87	45	Fear	2023-09-14	NaN
88	41	Fear	2023-09-13	NaN
89	30	Fear	2023-09-12	NaN
90	40	Fear	2023-09-11	NaN

91 rows × 4 columns

Figure vi

Second, min-max scaling was used to improve the dataset for sophisticated machine learning models. By standardizing the dataset as a whole, all features were scaled to the same value in this stage. Normalization and scaling together provide the groundwork for a comprehensive integration of sentiment analysis, improving the precision and dependability of our forecasts. These calculated adjustments improved our predictive models' emotional intelligence by addressing Twitter data access issues and introducing a thorough sentiment score via the alternative.me API.

Normalizing the data

```
[ ] scaler = MinMaxScaler()
```

```
[ ] final_dataset_n = scaler.fit_transform(final_dataset)
```

```
[ ] final_dataset
```

Date	closing_price	max_price	min_price	opening_price	sentiment
2023-09-11	25140.138899	25140.138899	25140.138899	25140.138899	40
2023-09-12	25979.215385	25133.303107	26288.334884	25133.303107	30
2023-09-13	26234.189828	25850.322300	26304.015465	25843.107805	41
2023-09-14	26615.532123	26222.013304	26751.796646	26193.031946	45
2023-09-15	26792.608825	26531.395566	26792.608825	26251.615685	45
...	...	...	...	...	...
2023-12-06	43634.904484	44105.944773	44123.756422	43613.258584	72
2023-12-07	43166.048404	43788.288808	43985.767023	43158.794425	72
2023-12-08	44252.835095	43270.119560	44385.975286	43174.140967	72
2023-12-09	43908.530003	44158.481190	44290.263279	43824.945022	73
2023-12-10	43866.175642	43714.935354	43968.397400	43634.411927	74

91 rows × 5 columns

Normalized dataset

```
[ ] final_dataset_n
```

```
array([[0.00000000e+00, 3.59302410e-04, 0.00000000e+00, 3.65713850e-04,
        2.22222222e-01],
       [4.37624614e-02, 0.00000000e+00, 5.96594485e-02, 0.00000000e+00,
        0.00000000e+00],
       [5.70607833e-02, 3.76879097e-02, 6.04742004e-02, 3.79744434e-02,
        2.44444444e-01],
```

Figure vii

The code begins by requesting the Fear and Greed Index from alternative.me through an API call, fetching sentiment data that reflects market emotions. The resulting data is then transformed into a structured DataFrame, allowing for seamless integration with our existing cryptocurrency dataset. The sentiment data, represented as scores on an emotional spectrum, is merged with our cryptocurrency dataset based on timestamp indices. The resulting dataset, aptly named final_dataset, is curated to include essential features such as closing price, max price, min price, opening price, and the newly integrated sentiment score. To ensure that all features are on a consistent scale suitable for machine learning models, the entire dataset is then normalized and scaled using the MinMaxScaler() from scikit-learn. The result is the final_dataset_n, a standardized data set ready for machine learning analysis.

1.3 Modeling

1.3.1 Attempted Models:

ARIMA : We initially tried out Auto-Regressive Integrated Moving Average model as a time series forecasting technique for predicting cryptocurrency prices. ARIMA is particularly well-suited for handling time-dependent data, making it a natural choice for this application.

```
ARIMA

[ ] X_train, X_test, y_train, y_test = train_test_split(features, target, test_size=0.2, random_state=42)

[ ] train_size = int(len(final_dataset_n) * 0.8)

[ ] train, test = final_dataset_n[:train_size], final_dataset_n[train_size:]

[ ] target_train = train[:, 0]
    target_test = test[:, 0]

[ ] model = ARIMA(target_train, order=(5,1,0))

    model_fit = model.fit()

[ ] forecast = model_fit.forecast(steps=len(test))

[ ] rmse = mean_squared_error(target_test, forecast, squared=False)
    print(f"RMSE: {rmse}")

    RMSE: 0.23469232071616947
```

Figure viii

The RMSE (Root Mean Squared Error) served as a performance metric, indicating the disparity between predicted and actual values. Despite its theoretical appropriateness, the ARIMA model exhibited a relatively high RMSE of 0.2347, indicating sub-optimal accuracy in predicting cryptocurrency price trends. This performance limitation prompted a reconsideration of the modeling strategy.

Random Forest Regressor: As an alternative to ARIMA, we also tried using the Random Forest model, an ensemble learning method leveraging multiple decision trees. The Random Forest Regressor yielded a significantly reduced RMSE of 0.0299, indicating a notable improvement in predictive accuracy compared to the ARIMA model.

Random Forest regressor

```
[ ] random_forest = RandomForestRegressor()

    random_forest.fit(X_train, y_train)
```

```
▸ RandomForestRegressor
```

```
[ ] rf_predictions = random_forest.predict(X_test)
```

```
[ ] rmse = mean_squared_error(y_test, rf_predictions, squared=False)
    print(rmse)

    0.02987323008225876
```

Figure ix

The Random Forest model exhibited a distinct improvement in prediction when compared to the ARIMA model, as indicated by the reduced RMSE.

Gradient Boosting Regressor: Another model we tried for comparison was Gradient Boosting, an ensemble learning technique building decision trees sequentially. The Gradient Boosting Regressor demonstrated enhanced accuracy, with an RMSE of 0.0288, surpassing the ARIMA model's performance.

Gradient Boosting Regressor

```
[ ] gradient_boosting = GradientBoostingRegressor()

    gradient_boosting.fit(X_train, y_train)
```

```
▾ GradientBoostingRegressor
GradientBoostingRegressor()
```

```
[ ] gb_predictions = gradient_boosting.predict(X_test)
```

```
[ ] rmse = mean_squared_error(y_test, gb_predictions, squared=False)

    print(rmse)

    0.028809367437606512
```

Figure x

Similar to the Random Forest model, Gradient Boosting demonstrated improved accuracy in predicting cryptocurrency price trends compared to the ARIMA model.

Linear Regression: We also explored the effectiveness of Linear regression as a predictive model. After training the Linear Regression model and evaluating its performance, we achieved an accuracy, with an RMSE of 0.01. With this accuracy, the model may not surpass some more complex models, it still showcases a notable level of performance and serves as a viable option. We decided to go with Logistic regression.

```
[128] scaler = MinMaxScaler()
      scaled_data = scaler.fit_transform(final_dataset)

      features = scaled_data[:, 1:]
      target = scaled_data[:, 0]

      X_train, X_test, y_train, y_test = train_test_split(features, target, test_size=0.2, random_state=42)

      model = LinearRegression()
      model.fit(X_train, y_train)

      predicted_normalized_price = model.predict(X_test)

      predicted_prices = scaler.inverse_transform(
          np.concatenate((predicted_normalized_price.reshape(-1, 1), X_test), axis=1))[:, 0]

[129] rmse = mean_squared_error(y_test, predicted_normalized_price, squared=False)
      rmse

      0.011017460166832546
```

Figure xi

1.3. 2 Implemented Model: Logistic Regression

After scaling the characteristics, a binary target variable is created based on whether it is expected that the closing price of the following day will be higher than the present one, preparing the dataset for predictive modeling. Next, 80% of the dataset is set aside for training, and 20% is set aside for testing. With the help of the training data set, a logistic regression model is created. The model's accuracy is then determined using the accuracy_score function after predictions are made on the testing data. An evaluation of the logistic regression model's predictability of an expected increase or reduction in the closing price is provided by the final accuracy %.

```
Logistic Regression

[73] features = scaled_data[:-1]  # Exclude the last day for prediction
     target = (final_dataset['closing_price'].shift(-1) > final_dataset['closing_price']).iloc[:-1]

[78] X_train, X_test, y_train, y_test = train_test_split(features, target, test_size=0.2, random_state=42)
     model = LogisticRegression()
     model.fit(X_train, y_train)

     ▾ LogisticRegression
     LogisticRegression()

[95] predictions = model.predict(X_test)

     accuracy = accuracy_score(y_test, predictions)

     print("Accuracy =", accuracy*100)
```

Figure xii

Our evaluation of various models highlighted the superiority of Logistic Regression in predicting cryptocurrency price trends with an accuracy of 77.77%

1.4 Web App and News Integration

1.4.1 Cryptocurrency Analysis Web App

Our project's essential element is the Cryptocurrency Analysis Web App. Designed with Streamlit, a Python package, the web application combines robust analytics with an easy-to-use interface. We can offer intricate market analysis in a way that is understandable to consumers of all skill levels, thanks to Streamlit's user-friendly design. With the use of interactive candlestick graphs and a logistic regression model, users of this web application may pick individual cryptocurrencies, examine past price trends, and forecast future price movements. Users can make informed decisions in a dynamic and fast-paced market environment by having access to the most recent market information thanks to real-time news data integration.
Features:

1. Real time price display: Users can select a cryptocurrency from the dropdown menu, instantly displaying the current price and the last update time. The 'Check' button allows users to refresh the data for real-time accuracy.

Figure xiii

2. Historical data visualization: The app presents historical data in two formats: a table showcasing the past five days information and a candlestick plot spanning the last 90 days. This visualization empowers users to track price trends and make informed decisions.

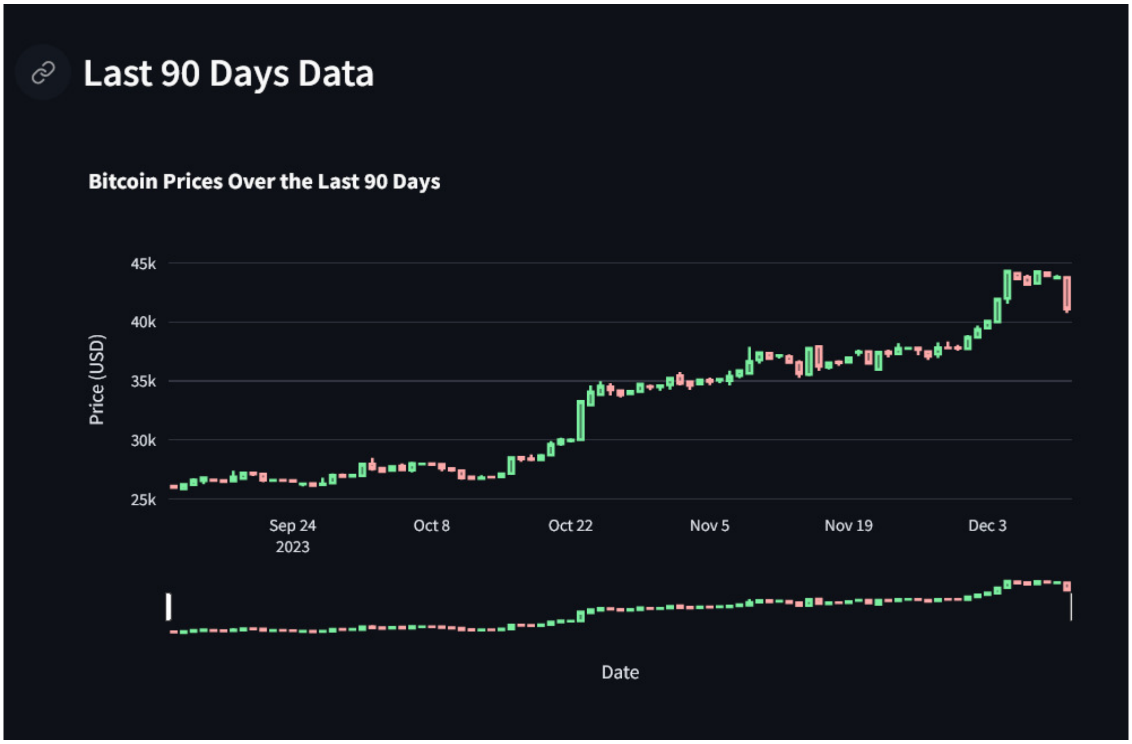


Figure xiv

3. Predictive Analysis: Users can predict the cryptocurrency's price change for different durations, such as the next day, one month, one year, and three years. Logistic regression is used for predictions based on historical data

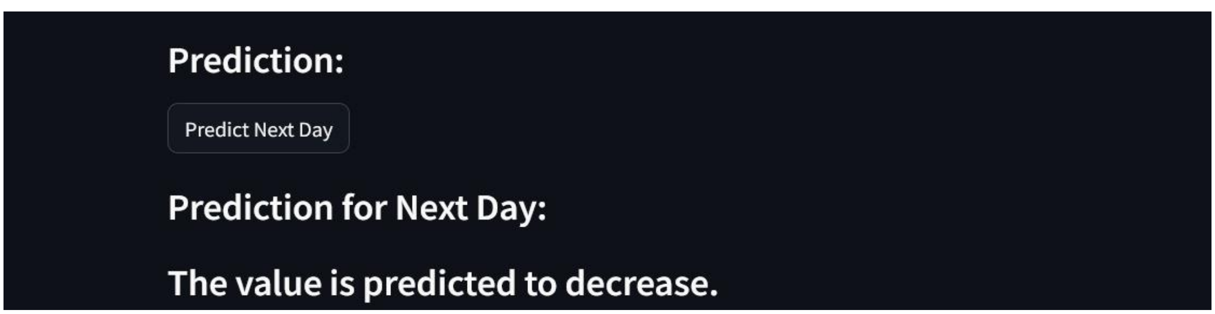


Figure xv

4. Sentiment analysis: The web app integrates the Fear and Greed Index to display sentiment values for the past five days. Users can gain insights into market sentiment trends.

Past 5 Days Sentiment Values

	value	value_classification
2023-12-11 00:00:00	74	Greedy
2023-12-10 00:00:00	74	Greedy
2023-12-09 00:00:00	73	Greedy
2023-12-08 00:00:00	72	Greedy
2023-12-07 00:00:00	72	Greedy

Figure xvi

5. Historic data based on the sentiment: Users can choose different time durations, like the next day, one month, three months, and one year, to view historical sentiment data.

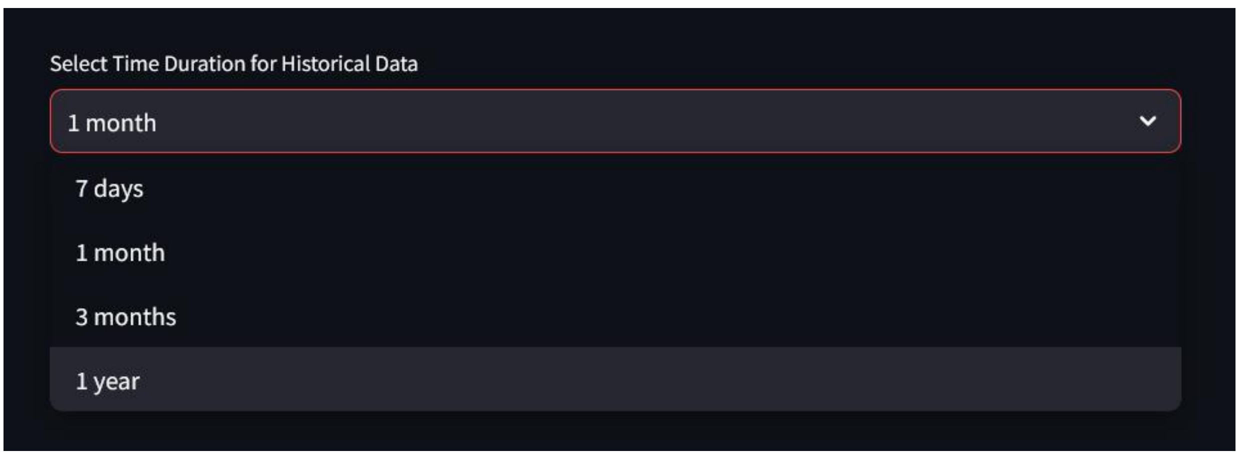


Figure xvii

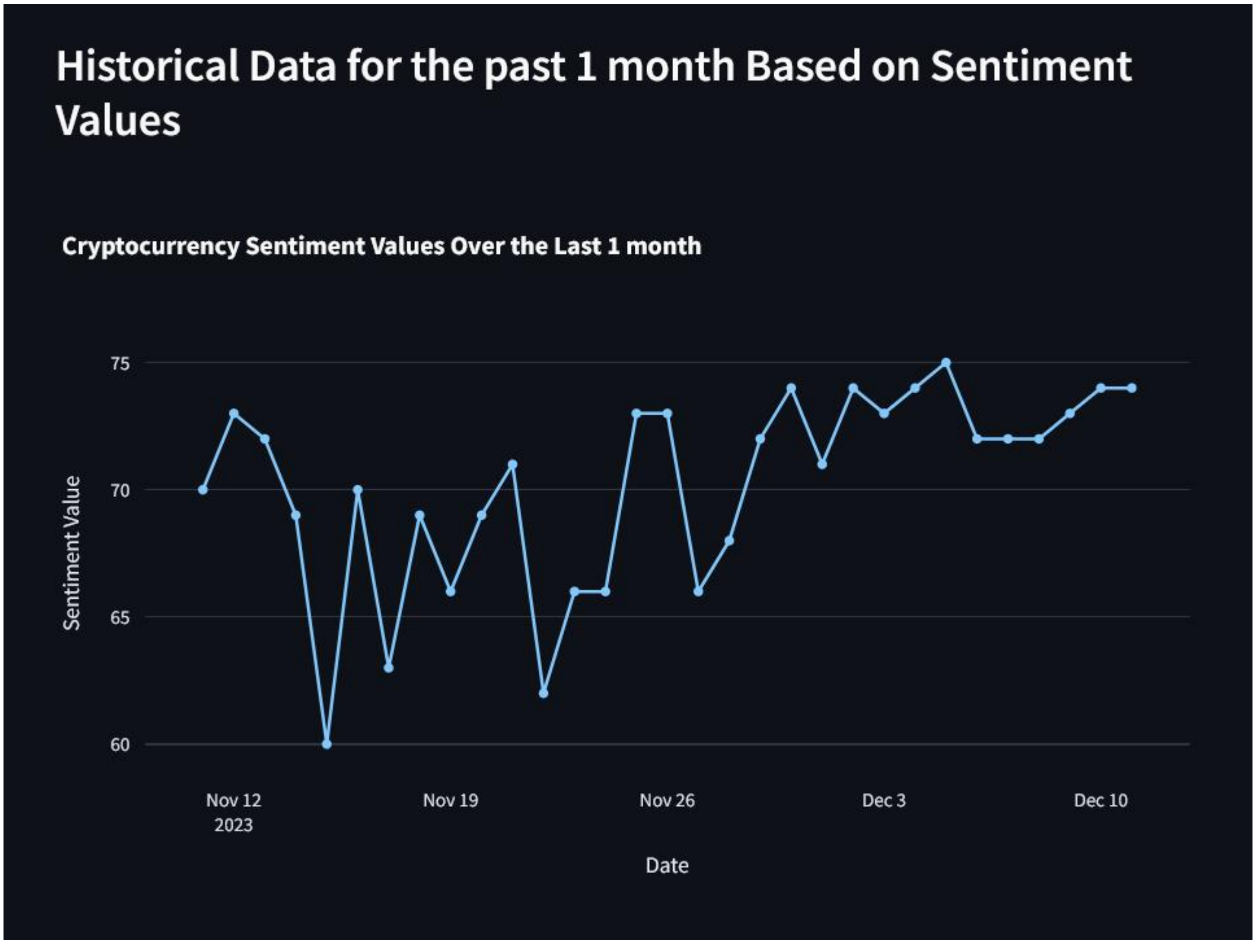


Figure xviii

1.4.2 News Integration

To elevate user experience and provide a more holistic market perspective, we integrated an additional layer of information through the newsapi.org API. This enhancement allows our users to access context-rich news articles directly related to their selected cryptocurrencies, bridging the divide between data-driven analysis and qualitative market factors. By integrating real-time news data, our app becomes more than just a predictive modeling tool; it evolves into a dynamic decision support system. This move ensures that users not only leverage data-driven predictions but also stay informed about external events, regulatory changes, and sentiment shifts that can significantly impact cryptocurrency prices. Leveraging the key features of the News API, such as extensive source coverage, ease of integration, customizable queries, real-time updates, consistent data format, and affordable pricing tiers, our integration enhances the app's functionality by delivering recent and relevant news related to users' selected cryptocurrencies. The News API, with its user-friendly interface and dynamic news updates, stands out as a reliable and versatile solution, reinforcing our app's capabilities in delivering timely and pertinent information to users navigating the intricacies of cryptocurrency investments.

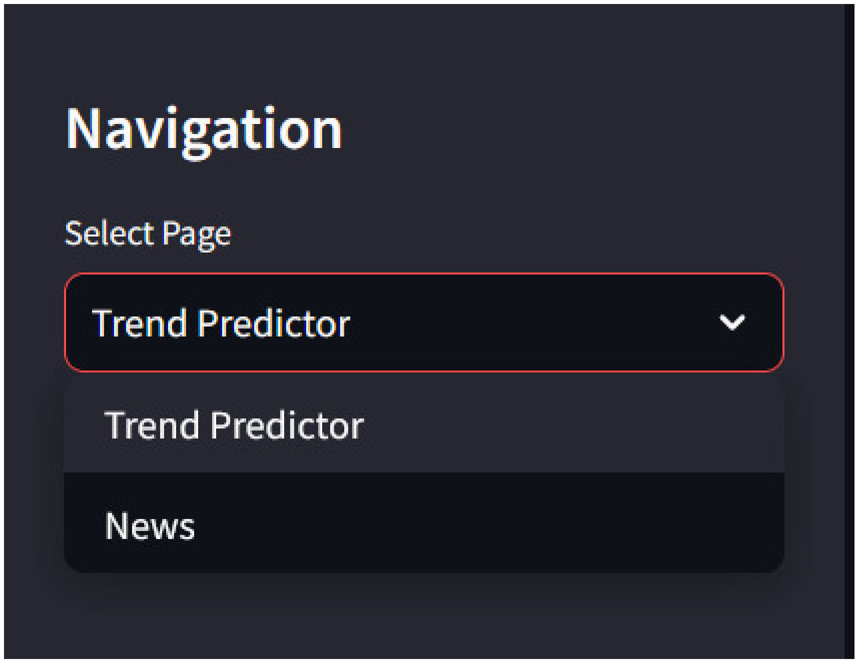


Figure xix

Features:

1. Coin Selection: Users can choose a cryptocurrency from the dropdown menu, and the app fetches and displays the latest news for the selected coin.

Figure xx

2. News display: Users can get the latest news articles with their titles, descriptions, sources, and a link to read more.

Figure xxi

2. Model Evaluation

2.1 Comparative Analysis

The cryptocurrency price prediction involved a strategic progression through various models, starting with ARIMA and transitioning to Random Forest, Gradient Boosting and Linear regression before ultimately settling on Logistic Regression. To implement each model, the Scikit-learn library was used. Modern models with good performance and versatility are available in a toolbox provided by Scikit-learn. Because of their great accessibility and data availability, the digital currency markets can be a topic with a lot of possibilities to study on financial time series problems (Valencia et al., 2019). Random Forest, with its ensemble learning methodology, presented a substantial leap forward in accuracy, followed by Gradient Boosting and Linear regression, which further fine-tuned our predictive models. However, recognizing the volatile and multifaceted nature of cryptocurrency markets, logistic Regression emerged as a robust choice, showcasing adaptability to the intricate dynamics of price trends. Despite these advancements, achieving pinpoint accuracy in this dynamic landscape remains elusive. Thus, our approach embraces a holistic strategy, incorporating supplementary tools such as sentiment analysis and news integration. These additions contribute to a comprehensive understanding, acknowledging the challenges posed by external factors like market sentiment, regulatory shifts, and macroeconomic trends. Our journey reflects a continuous quest for improvement and innovation in the realm of cryptocurrency price prediction, emphasizing the importance of dynamic adaptation and nuanced insights for enhanced predictive accuracy in this ever-evolving landscape.

2.2 Limitations

The real-time news data integration empowers users with timely and accurate information, a crucial aspect in the volatile realm of cryptocurrency trading. However, the model is not without its limitations. The inherent complexities of cryptocurrency markets, characterized by sudden price fluctuations and external influences, pose challenges to achieving precise predictions.

3. Conclusion

In our project, we aimed to find how cryptocurrency prices might change. We tested various methods like ARIMA, Random Forest, Gradient Boosting, Linear regression and eventually chose Logistic Regression. The integration of the PyCoinGecko API helped our model to efficiently forecast general trends, maintaining relevance in the dynamic cryptocurrency realm. Acknowledging the impact of external factors and market sentiment, we integrated sentiment analysis and news data, providing a nuanced perspective on cryptocurrency price movements. While Logistic Regression has proven its prowess, it's crucial to emphasize that our project is a tool for predicting trends, and users must conduct their own research before making investment decisions.

We transformed our project into a user-friendly experience by leveraging Streamlit for deployment. Users can explore the intricacies of predicting crypto price trends, and Streamlit's effortless interface makes forecasting accessible to everyone.

4. Next Steps and Future Enhancements

In the ever-changing world of predicting cryptocurrency markets. Here are the next things we plan to do and the improvements we want to make.

4.1 Model Refinement and Continuous Adaptation:

We worked on making our cryptocurrency price prediction models better by tweaking existing ones and trying out new approaches. This involved adjusting settings and exploring different methods to deal with the unique challenges in the cryptocurrency world. Our goal was to improve the accuracy of our models and make sure they can handle the ever-changing market. At the same time, we understand that cryptocurrency markets are always changing. So, we've made a commitment to keep an eye on things, regularly check how our models are doing, and make changes as needed. This way, we can make sure our models stay helpful and provide valuable insights to people dealing with the ups and downs of cryptocurrency investments.

4.2 Alternative Sentiment Analysis:

Recognizing the pivotal role sentiment analysis plays in our forecasting models, we are committed to exploring alternative solutions. Our focus is on identifying and integrating advanced sentiment analysis techniques that can further enhance accuracy and robustness in capturing market sentiment. By diversifying our approach to sentiment analysis, we aim to fortify our models against potential limitations and better capture the nuanced shifts in investor sentiment that influence cryptocurrency price movements.

5. References

API's:

1. CoinGecko API: https://www.coingecko.com/en/api
2. alternative.me API: https://alternative.me/crypto/
3. newsapi.org API: https://newsapi.org/

References:

1. Wimalagunaratne, M., & Poravi, G. (2018). A Predictive Model for the Global Cryptocurrency Market: A Holistic Approach to Predicting Cryptocurrency Prices. In 2018 8th International Conference on Intelligent Systems, Modelling and Simulation (ISMS) (pp. 78-83). IEEE. DOI: 10.1109/ISMS.2018.00024.
2. Valencia, F., Gómez-Espinosa, A., & Valdés-Aguirre, B. (2019). Price Movement Prediction of Cryptocurrencies Using Sentiment Analysis and Machine Learning. Entropy, 21, 589. DOI: 10.3390/e21060589.
3. Felizardo, L., Oliveira, R., Del-Moral-Hernandez, E., & Cozman, F. (2019). Comparative study of Bitcoin price prediction using WaveNets, Recurrent Neural Networks and other Machine Learning Methods. In 2019 6th International Conference on Behavioral, Economic and Socio-Cultural Computing (BESC) (pp. 1-6). IEEE. DOI: 10.1109/BESC48373.2019.8963009.