Documentation

The aim of this project is to predict the future cryptocurrency price difference based on recent historical data using both engineered trading metrics and machine learning algorithm . Accurate predictions can aid traders and investors by offering insights into potential price movements

Objective:

1. **Data Retrieval and Preprocessing** : Retrieve historical Bitcoin prices through an API and preprocess it to create relevant trading metrics.
2. **Feature Engineering** : Generate Specific features based on historical prices ,such as number of days such as last high and low prices , and the percentage difference from those values
3. **Model Training and Tuning** : Train a machine learning model to predict two target metrics indicating future price changes.
4. **Evaluation** : Evaluate the model accuracy on metrics such as Root mean squared

1. **Data Retrieval and Preprocessing**
   API Data Retrieval : We start by using a CoinMarketCap API to obtain historical Bitcoin data, which includes price data for each timestamp . API- based retrieval ensures that we have the latest data and allow for regular updates to improve prediction accuracy over time.

2. **Feature Engineering**

   The following four metrics are derived from recent historical data over the past 7 days. The api retrieval only provided four features(Open, High, Low , Close)

   - **Days_Since_High_Last_Days:** Count days since the most recent high price within the past 7 days
   - **%_Diff_From_High_Last_Days:** Measures the percentage difference between the current price and the highest price within the past 7 days
   - **Days_Since_Low_Last_Days:** Counts days since the most recent low price within the past 7 days
   - **%_Diff_From_Low_Last_Days**: Measures the percentage difference between the current price and the lowest price within the past 7 days

These features are highly informative as they capture recent trends and market behavior ,which are crucial for predicting short term movements

Target Variable:

We aim to predict two target metrics that provide insights into potential future price movements over next 5 days

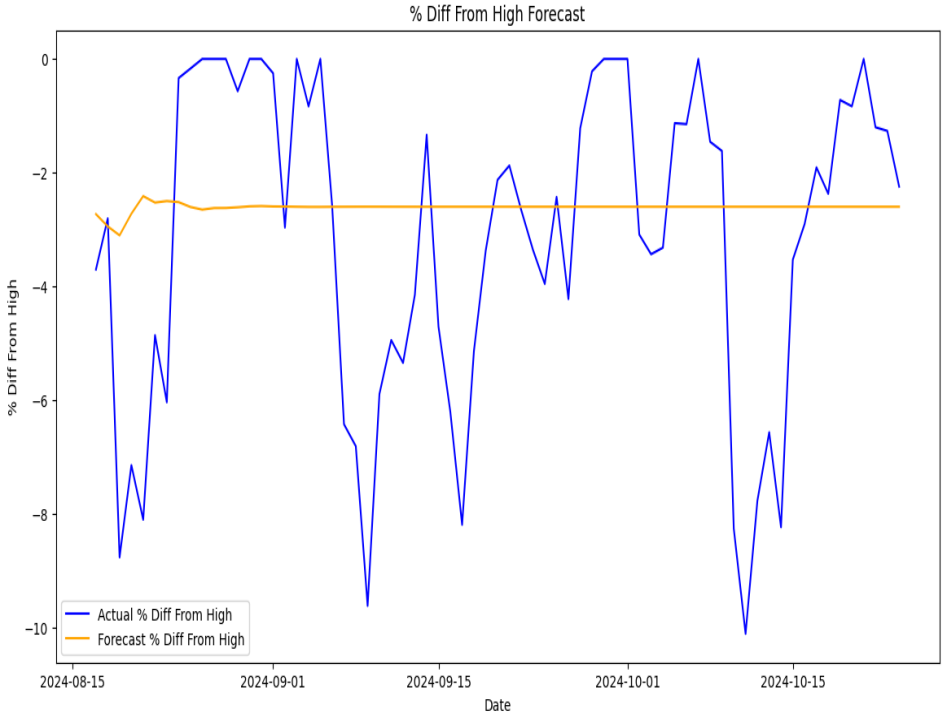
- **%Diff_From_High_Next_5_days**: The percentage difference from the highest price within the next 5 days
- **%Diff_From_Low_Next_5_days:** The percentage difference from the lowest price within the next 5 days

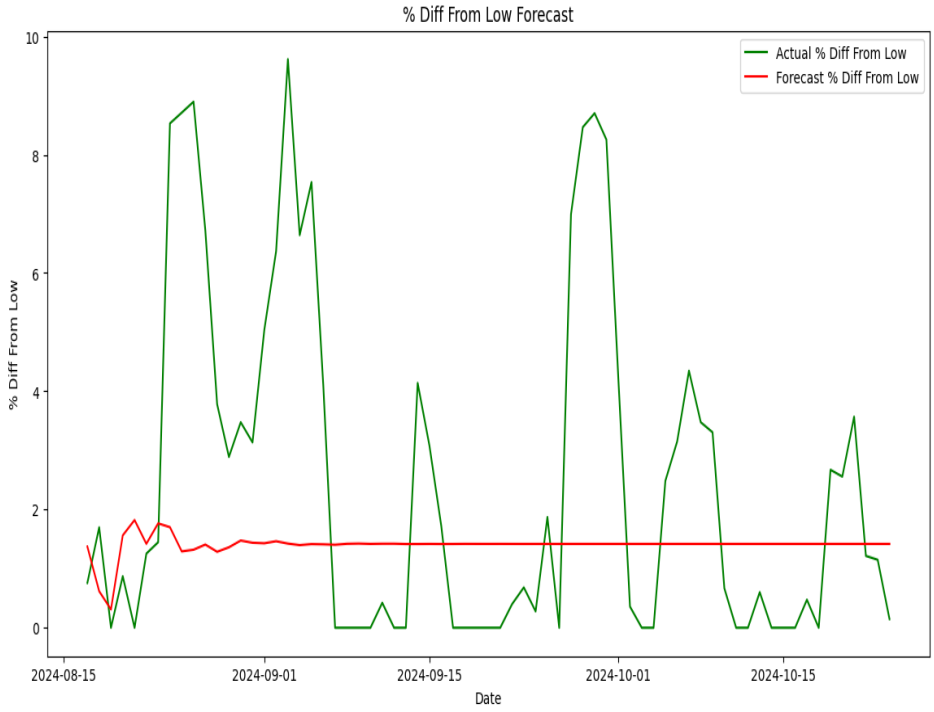
3. **Model Training And Tuning**

   Model Selection

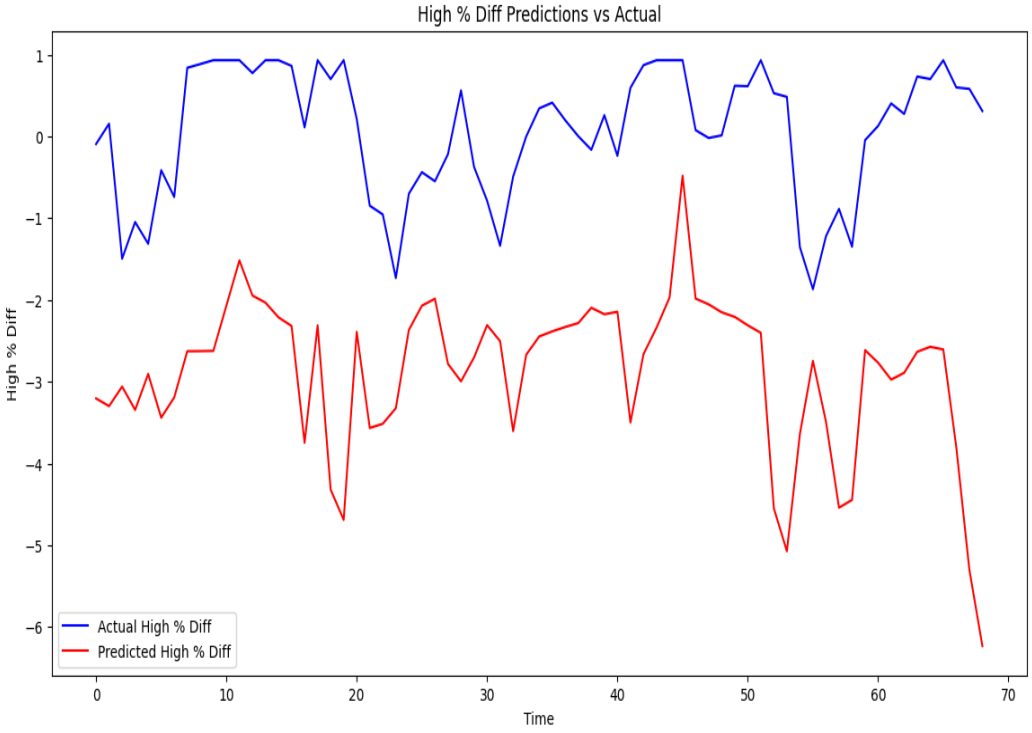
We selected RandomForestRegressor due to its effectiveness in capturing complex relationship and its resistance to overfitting, particularly when handling noisy financial data. We tried out different models such as ARIMA , LSTM . The ensemble method is ideal for identifying trends without relying too heavily on linear patterns , which is essential for time series data

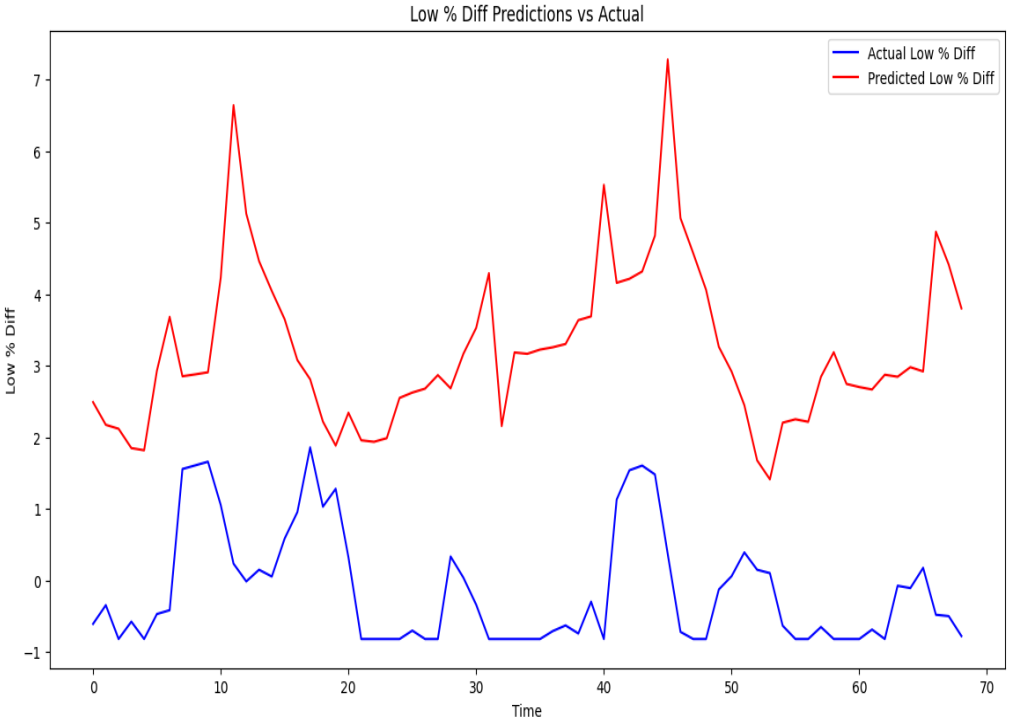
ARIMA Model :


% Diff From High Forecast
% Diff From High
Date
Actual % Diff From High
Forecast % Diff From High
0
−2
−4
−6
−8
−10
2024-08-15
2024-09-01
2024-09-15
2024-10-01
2024-10-15

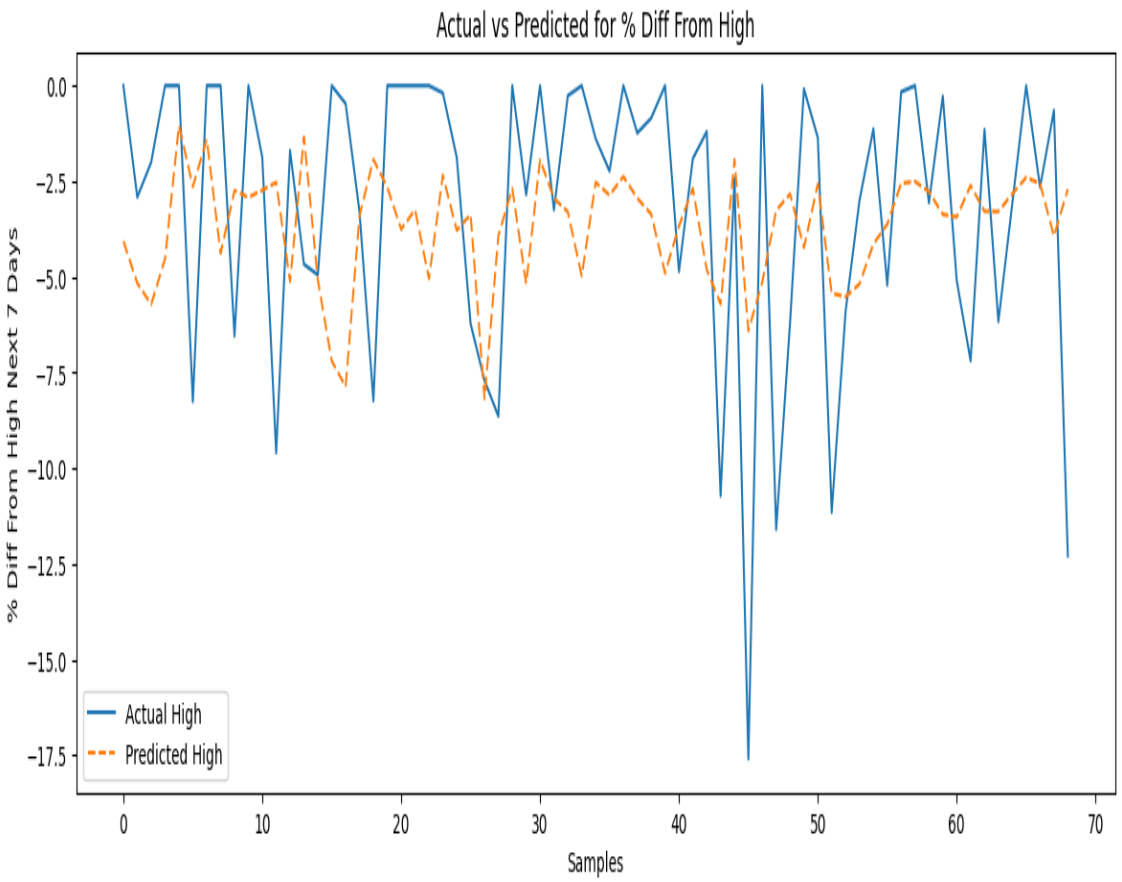

% Diff From Low Forecast
% Diff From Low
Date
Actual % Diff From Low
Forecast % Diff From Low
10
8
6
4
2
0
2024-08-15
2024-09-01
2024-09-15
2024-10-01
2024-10-15

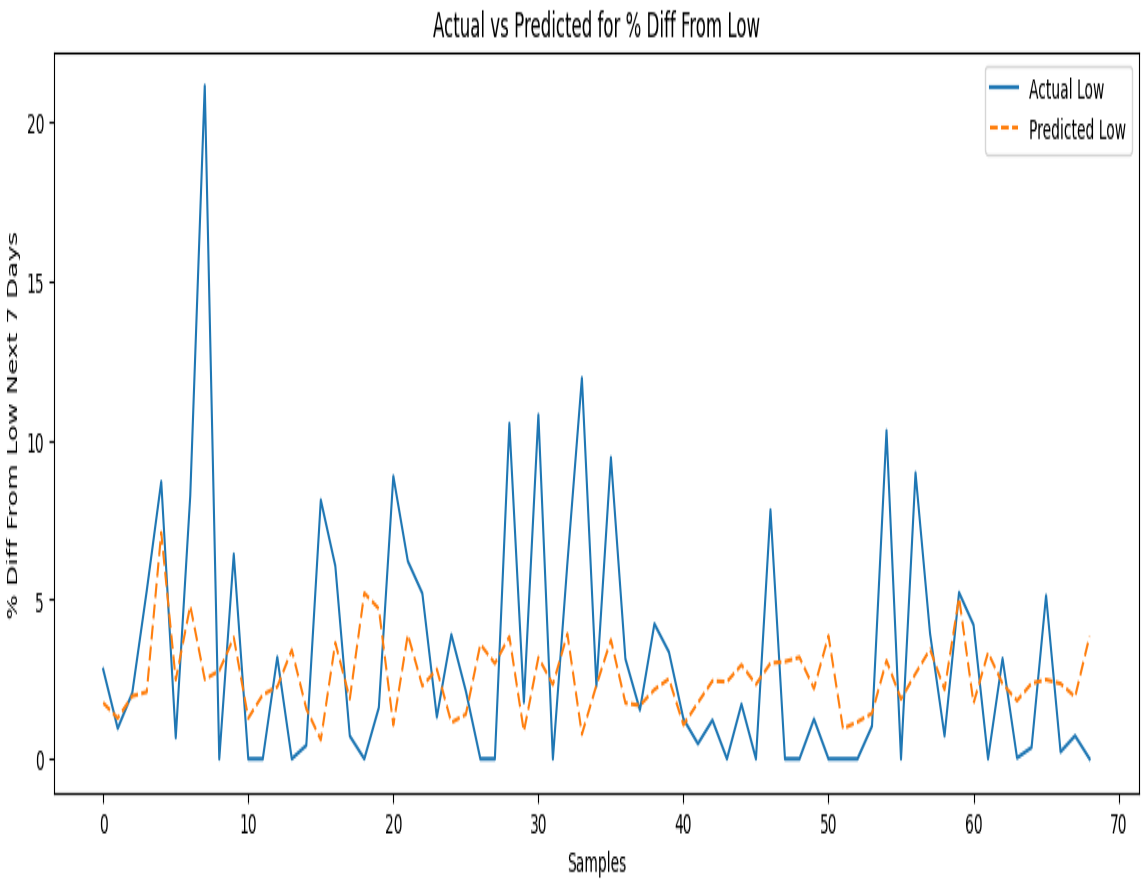
LSTM Model:





RandomForestRegressor Model :


Actual vs Predicted for % Diff From High
% Diff From High Next 7 Days
0.0
−2.5
−5.0
−7.5
−10.0
−12.5
−15.0
−17.5
0
10
20
30
40
50
60
70
Samples
Actual High
Predicted High



Hyperparameter Tuning

We applied GridSearchCV to perform hyperparameter tuning in key parameters like n_estimator , max_depth, and max_features . This allowed us to find an optimal configuration of model, balancing their bias variance tradeoff and improving precision

Training the Model

After tuning , the model was trained on 80% of the data and remaining 20 % hold back for testing . This split ensured that the model was evaluated on unseen data to simulate the real world performance and prevent overfitting.

4. **Evaluation :**
   The model accuracy was addressed using RMSE(Root Mean Squared Error),a common metric for regression models that measures the average magnitude of prediction of errors .

5. Challenges and Solutions
   - Handling Data gaps and missing values : The initial dataset had some missing values due to market inactivity on certain dates. These rows were removed to maintain a clean dataset , as filling them could introduce bias or misleading trends.
   - Model Selection : It was difficult to choose between models. Future work include experimenting with other machine learning models like XGBoost or neural network to capture dependencies more effectively

Tools Used : Google Colab , Microsoft Excel ,VSCode

Libraries used : Scikit Learn, Numpy , Pandas , Matplotlib