CRYPTO PRICE PREDICTION

Forecasting next day crypto close price

Sumit Redekar

TABLE OF CONTENT

INTRODUCTION

In Crypto Market Prediction, our aim is to build an efficient Machine Learning model to predict the future value of the given crypto. The recent trend in crypto market prediction technologies is the use of machine learning which makes predictions based on the values of current crypto market indices by training on their previous values. Our project focuses on the use of Regression models and the LSTM model to predict crypto values. Factors considered are open, close, low, high, and volume. Data will be taken from CoinMarketCap with the help of a python scraper in the .csv format and as for the data cleaning and handling missing values, we will use Pandas and NumPy. Data visualization will be done with matplotlib and seaborn. Machine Learning modeling will be done with scikit-learn and Deep Learning with TensorFlow/Keras. Our machine-learning model will be presented to retail investors with a third-party web app with the help of Streamlit.

LANGUAGE/LIBRARY USED

For this whole project we have used Python Language and Python Machine Learning libraries as follows:

API/ Web Scraper: CryptoCmd python scraper scraped the data from CoinMarketCap

Data Wrangling: Pandas, Numpy

Data Visualisation: Matplotlib, Seaborn

Data Modeling: Scikit: Learn, TensorFlow, Keras

Database: SQL

Webapp: Streamlit

IDE: Jupyter Notebook/ PyCharm

STEP-BY-STEP METHOD TO TRAIN AND DEPLOY THE MODEL

With the help of Machine Learning, we will build an efficient and robust ML model to predict the next day's price of the selected crypto. Our model will fit the last 5 years of historical data and while predicting it will look for the past 3 months of the data and will predict the next day's price.

It contains different steps that are important while building the model.

1. **Data Gathering**: Data gathering involves gathering data for analyzing and training the model. Our data should be correct in every aspect as it will predict and analyze the trends in historical prices.
2. **Data Wrangling:** Data wrangling involves cleaning the data. We can't train the model with having string data type or the data with missing values. So we should always clean our data before fitting the model.
3. **Data Analyzing**: Data Analyzing includes analyzing historical stock market prices and gaining hidden insights from the data.
4. **Splitting the data**: Now our data is ready for training, but before that there is one essential step is to split our data into training and testing data. We will use our training data to train and test data to evaluate i.e. to check how our model performs on unseen data. For that, we have found that it is best practice to split our data into an 8:2 ratio (80% for training and 20% for testing).
5. **Training the model**: The main important part of this project is to train the efficient machine learning model. With the help of Python and python libraries, we will train and compare our various models.
6. **Evaluating/Testing the model**: After training the model, the next step is to check the accuracy of our model on the test data and to get our base model which performs well on unseen data.
7. **Hypertuning the model:** In this step, we will hyper-tune some parameters of the base model and we will try to increase our accuracy. In this step, we will get our best and final model.
8. **Deploying the model:** After so many steps we have actually found our best model. So in this step. we will deploy this model and create our web app. It will serve as a third-party web app for retail investors.

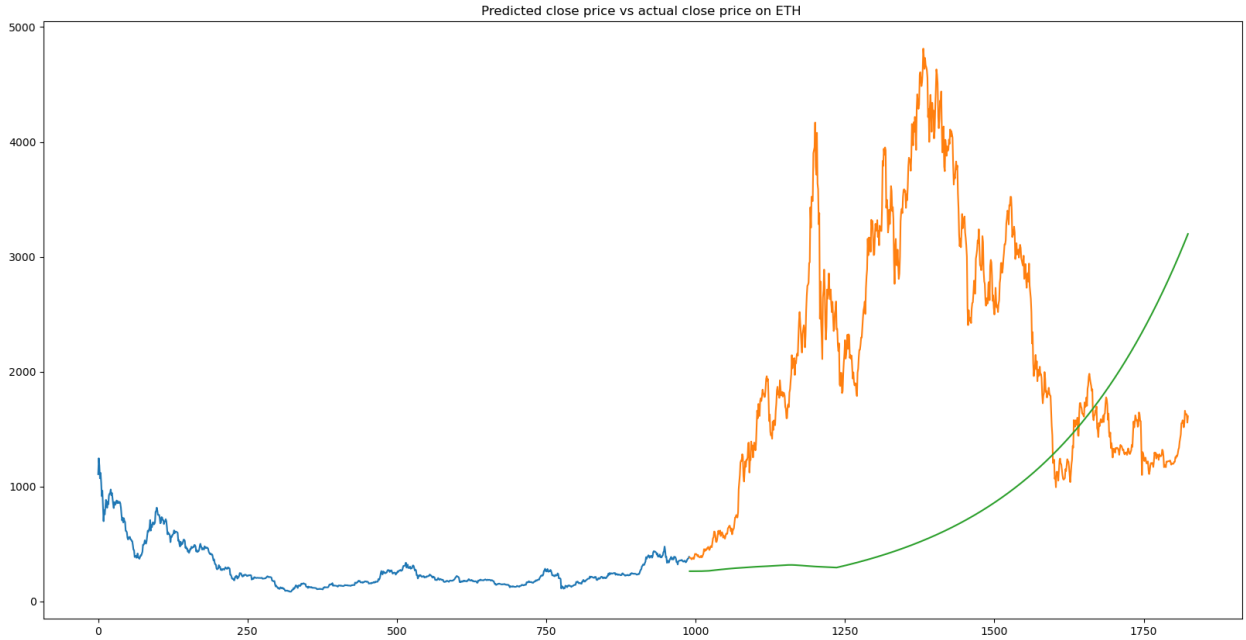
MACHINE LEARNING MODEL DEFINITION & COMPARISON

As the main focus of the project is to develop an efficient and robust machine learning model to predict the next day's price so we have trained and fitted 3 different traditional Machine Learning and 1 deep learning model that is LSTM (Long Short Term Memory). In this project, we have used three other Machine Learning models that are Linear Regression, K-Nearest Neighbors, and Moving Average. We have also used a deep learning model which is the LSTM model to fit the historical data. After fitting the model we plotted the actual and predicted values and used the Accuracy & RMSE metric so we could decide which model to choose.

(Note: All these graphs are based on Ethereum)

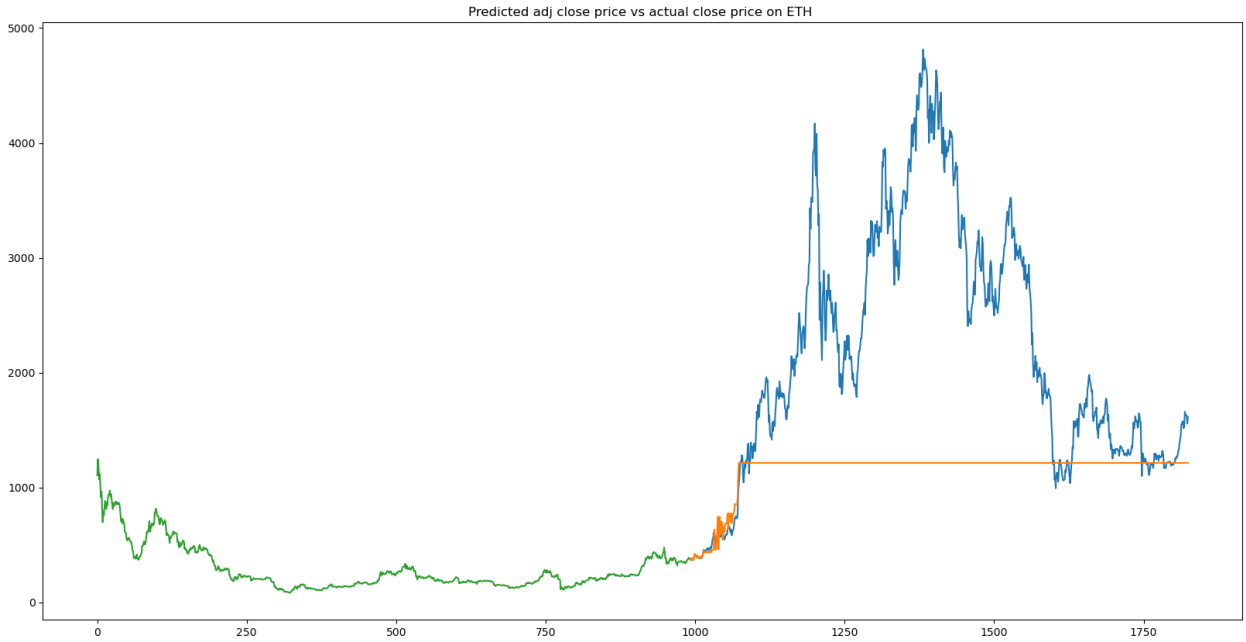
1. **MOVING AVERAGE MODEL**

   We have started with a simple and powerful model which is the Moving Average model. This model will fit the past 5 years of data and predict the prices on the validation dataset. The predicted closing price for each day will be the average of a set of previously observed values. So this model has not performed well and RMSE was 1962, so we will continue with another model.



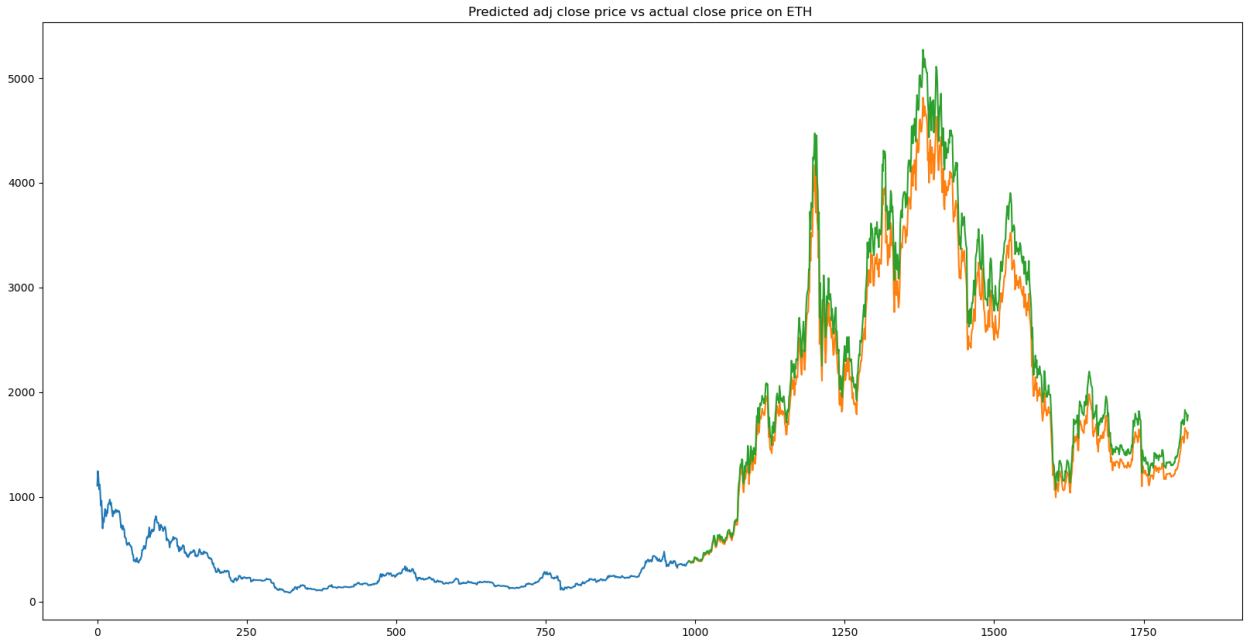
2. LINEAR REGRESSION MODEL

The most basic machine learning algorithm that can be implemented on this data is linear regression. The linear regression model returns an equation that determines the relationship between the independent variables and the dependent variable. This model is far better than the Moving Average and the RMSE was 232. Better than the Moving Average but we cannot use this in real life.



3. K-NEAREST NEIGHBOR MODEL

Another interesting ML algorithm that one can use here is kNN (k nearest neighbors). Based on the independent variables, KNN finds the similarity between new data points and old data points. This model has recognized some trends in the prices of stocks but still, it's not efficient to use. The RMSE of this model is 1432, this is worst than the moving average and linear regression.



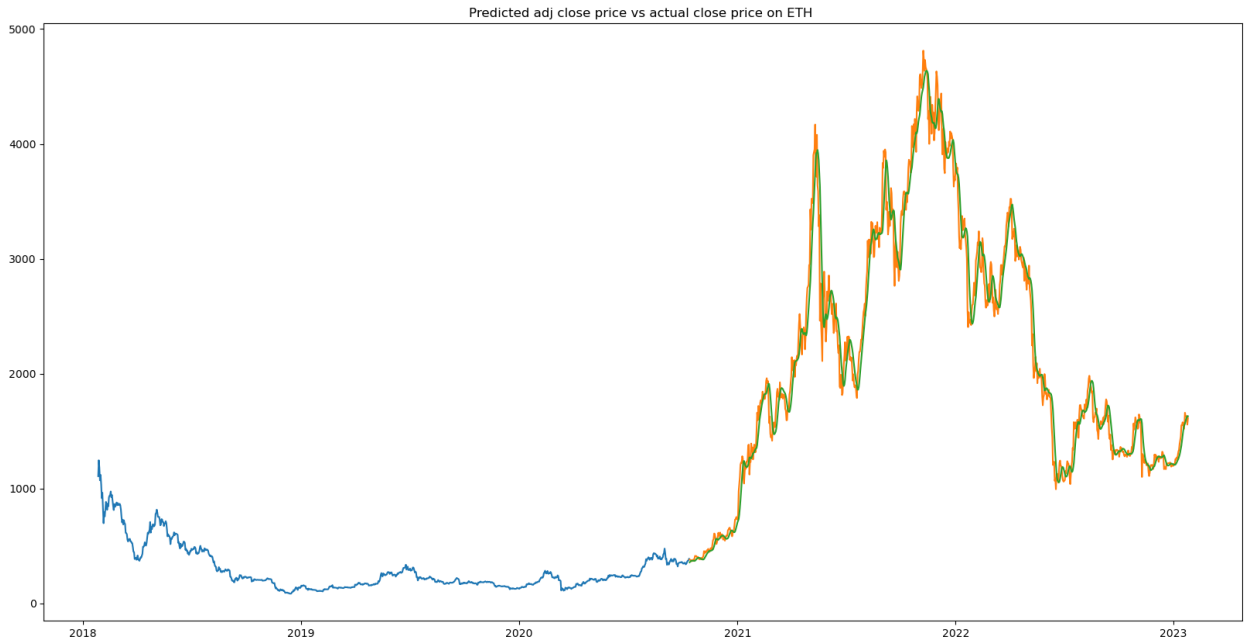
4. LSTM MODEL

Our last and most powerful model is this deep learning model. LSTMs are widely used for sequence prediction problems and have proven to be extremely effective. The reason they work so well is that LSTM is able to store past information that is important and forget the information that is not.

LSTM has three gates:

a. The input gate: The input gate adds information to the cell state
b. The forget gate: It removes the information that is no longer required by model
c. The output gate: The output Gate at LSTM selects the information to be shown as output This deep learning model has done some work here.

It has recognized all the potential upward and downtrends in stock prices. It is much more efficient compared to all other different ML models and the RMSE of the model is 180.



METRICS REPORT

1. On ETH

MODEL	ACCURACY (%)	ERROR (RMSE)
Moving Average	-227	1962
K Nearest Neighbor	-74	1432
Linear Regression	95	232
LSTM	97	180

2. On BTC

MODEL	ACCURACY (%)	ERROR (RMSE)
Moving Average	-562	37513
K Nearest Neighbor	-242	26988
Linear Regression	98	1624
LSTM	87	4557

RESULTS & CONCLUSION

After looking at all these 4 different model metrics, we can clearly see that the LSTM model and Linear Regression model outperforms all the other model. So in our web app, we can either use the Linear regression model or LSTM.

NOTE FOR CLIENT

If you loved the project, I am always up to discuss this project or to improve it more in any way. This project already has a fully-fledged version with features like Technical Indicators, Screener, and Pattern Recognition. I will attach the demo video of both versions i.e. forecasting one and the fully-fledged one.