ETHEREUM PRICE PREDICTION

A Course Project report submitted

in partial fulfillment of requirement for the award of degree

BACHELOR OF TECHNOLOGY

in

COMPUTER SCIENCE AND ENGINEERING

by

S.VIVEK KRISHNA	**(2103A51537)**
V.SAI SHASHANK	**(2103A51287)**
SK.SIRAJ	**(2103A51033)**

Under the guidance of

Mr. D. RAMESH

Assistant Professor, Department of CSE.

Department of Computer Science and Artificial Intelligence

Department of Computer Science and Artificial Intelligence

CERTIFICATE

This is to certify that project entitled **"ETHEREUM PRICE PREDICTION**" is the bonafied work carried out by **S.VIVEK KRISHNA, V.SAI SHASHANK, SK.SIRAJ** as a Course Project for the partial fulfillment to award the degree **BACHELOR OF TECHNOLOGY** in **COMPUTER SCIENCE AND ENGINEERING** during the academic year 2022-2023 under our guidance and Supervision.

Mr. D. RAMESH
Asst. Professor,
S R University,
Ananthasagar ,Warangal

Dr. M. Sheshikala
Assoc. Prof .& HOD (CSE)
S R University,
Ananthasagar ,Warangal

ACKNOWLEDGEMENT

We express our thanks to Course co-coordinator **Mr. D.Ramesh, Asst. Prof.** for guiding us from the beginning through the end of the Course Project. We express our gratitude to Head of the department CS&AI, **Dr. M.Sheshikala, Associate Professor** for encouragement, support and insightful suggestions. We truly value their consistent feedback on our progress, which was always constructive and encouraging and ultimately drove us to the right direction.

We wish to take this opportunity to express our sincere gratitude and deep sense of respect to our beloved Dean, School of Computer Science and Artificial Intelligence, **Dr C. V. Guru Rao**, for his continuous support and guidance to complete this project in the institute.

Finally, we express our thanks to all the teaching and non-teaching staff of the department for their suggestions and timely support.

ABSTRACT

- Ethereum is a blockchain-based decentralized platform that enables the creation and execution of smart contracts and decentralized applications (dApps).
- It was created in 2015 by Vitalik Buterin and has since become one of the most widely used and popular blockchain platforms. Ethereum uses a native cryptocurrency called Ether (ETH) as its fuel for transactions and computational operations on the network.
- One of Ethereum's unique features is its ability to create and execute complex smart contracts, which can automate various financial and non-financial applications. Additionally, Ethereum is constantly evolving, with developers working on upgrades to improve scalability and enhance functionality.
- Overall, Ethereum has opened up a new era of decentralized technology, paving the way for a more transparent and decentralized future.

Table of Contents

Chapter No.	Title	Page No.
1.	Introduction	
1.1.	Overview	1
1.2.	Problem Statement	1
1.3.	Existing system	1
1.4.	Proposed system	2
1.5.	Objectives	2
1.6.	Architecture	2
2.	Literature survey	
2.1.1.	Document the survey done by you	3
3.	Data pre-processing	
3.1.	Dataset description	4
3.2.	Data cleaning	4
3.3.	Data Visualization	5
4.	Methodology	
4.1	Procedure to solve the given problem	7
4.2	Model architecture	9
4.3.	Software description	9
5.	Results and discussion	10
6.	Conclusion and future scope	15
7.	References	15

INTRODUCTION

1.1.OVERVIEW

Ethereum is a decentralized, open-source blockchain with smart contract functionality.
Ether is the native cryptocurrency of the platform. Among cryptocurrencies, ether is second only to bitcoin in market capitalization.
Ethereum was conceived in 2013 by programmer Vitalik Buterin. In 2014, development work began and was crowdfunded, and the network went live on 30 July 2015.
Ethereum allows anyone to deploy permanent and immutable decentralized applications onto it, with which users can interact. Decentralized finance(DeFi) applications provide a broad array of financial services without the need for typical financial
intermediaries like brokerages, exchanges, or banks, such as allowing cryptocurrency users to borrow against their holdings or lend them out for interest.

1.2. PROBLEM STATEMENT

To develop a model which can help us to predict the price of the crypto currency used (in this case: Ethereum), with low error rate and a high precision of accuracy. The model will not tell the future, but it might forecast the general trend and the direction to expect the prices to move.

1.3. EXISTING SYSTEM

Firstly, we collect the data set from the online source: Yahoo. The data set represents the Ethereum price in United States Dollars (USD). The dataset includes all the information about bitcoin prices from 18 January,2022 to 18 January,2023. The second step involves filtering and cleaning the data set. This involves removing all the incomplete data from the rows. It also involves filtering out unnecessary features present in the data collected.

1.4. PROPOSED SYSTEM

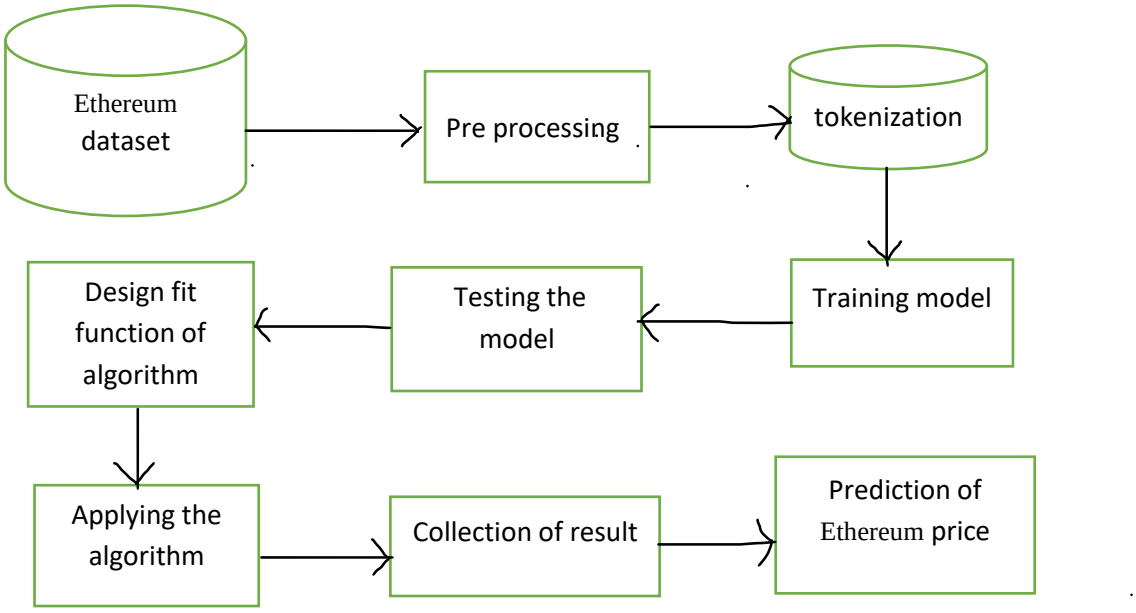
The step is training, followed by testing the dataset. We train our model, using the algorithm and the features taken into account to assist our model, to predict the future price of the crypto currency. Moving on to the testing part, we test the data to measure the accuracy of the algorithm that our model is using to predict the price of the Ethereum.

1.5. OBJECTIVES

The main objective of this research is to develop a model which can help us to predict the price of the crypto currency used (in this case: Ethereum), with low error rate and a high precision of accuracy. The model will not tell the future, but it might forecast the general trend and the direction to expect the prices to move

1.7. ARCHITECTURE

The architecture of the proposed system is as displayed in the figure below. The major components of the architecture are as follows: Ethereum dataset, preprocessing, tokenization, training the model, test the model, design fitness function, application of algorithm, results collection and prediction of Ethereum disease.



2. LITERATURE SURVEY

Here are some of the key papers and articles that provide a literature survey on Ethereum price prediction:

1. "Ethereum Price Prediction with Sentiment Analysis for Smart Trading" by R. Tandon et al. (2021)

This paper provides a literature survey on various techniques used for Ethereum price prediction, including machine learning and sentiment analysis. The authors then propose a new method that combines both techniques to predict Ethereum prices with higher accuracy.

2. "Bitcoin and Ethereum price forecast using historical trading volumes and network parameters" by V. Kovanen et al. (2018)

This paper explores the use of trading volumes and network parameters to forecast Bitcoin and Ethereum prices. The authors conduct an empirical study using historical data and show that network parameters can be used to predict price movements.

3. "Forecasting Cryptocurrency Prices with Machine Learning" by C. Ge and C. Lai (2018)

This paper provides a literature survey on various machine learning techniques used for cryptocurrency price prediction, including linear regression, random forest, and LSTM neural networks. The authors then propose a new method that combines LSTM with wavelet decomposition to predict Ethereum prices.

Overall, these papers and articles provide a comprehensive survey of the different techniques used for Ethereum price prediction, including both traditional machine learning and deep learning techniques. The literature survey highlights the importance of incorporating various data sources, including network parameters and sentiment analysis, for accurate price prediction.

3.DATA PRE-PROCESSING

3.1.1 DATASET DESCRIPTION

Sno	Attributes	Description
1.	OPEN	The opening price of the time period.
2.	HIGH	The highest price of the time period.
3.	LOW	The lowest price of the time period.
4.	CLOSE	The closing price of the time period.
5.	VOLUME BTC	This is the volume in the transacted ccy. BTC-Ethereum

3.2 DATA CLEANING

Data cleansing is a valuable process that can help companies save time and increase their efficiency. Data cleansing software tools are used by various organizations to remove duplicate data, fix and amend badly-formatted, incorrect and amend incomplete data from marketing lists, databases and CRM's. Data quality has become an important issue. This issue becomes more and more important in medicine area, where the need for effective decision making is high. In this context, the need for data cleaning to improve data quality is becoming crucial. Duplicate records elimination is a challenging data cleansing task. In this paper, we present a duplicate records elimination approach to improve the quality of data. We propose a deep learning-based approach for duplicate records detection using a sentence embeddings model. Also, we propose an algorithm for duplicated records correction. Then we apply the proposed duplicate records elimination approach to analyses the effect of data cleaning on the quality of decisions.

3.4 DATA VISUALISATION

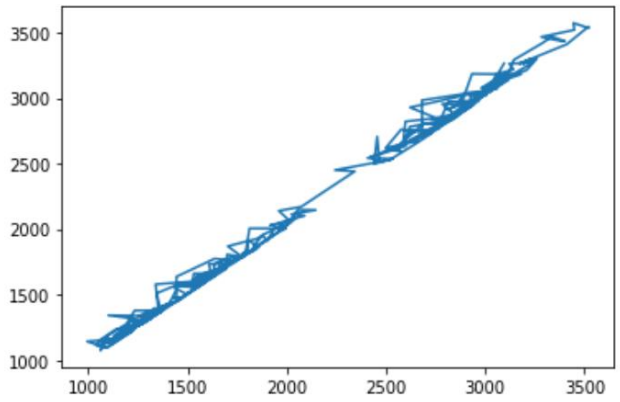
The historical Ethereum data set contains six feature variables and one target variable output.

DATASET

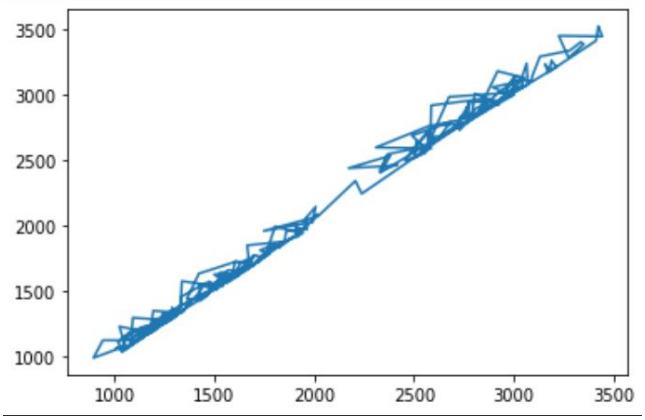
	A	B	C	D	E	F
1	Open	High	Low	Close	Adj_Close	Volume
2	3095.271729	3265.336914	3000.908203	3001.120117	3001.120117	10645922764
3	3002.956787	3029.081055	2496.812988	2557.931641	2557.931641	26796291874
4	2561.145264	2615.247314	2330.247314	2405.181152	2405.181152	27369692036
5	2406.924316	2542.144775	2381.515137	2535.039063	2535.039063	16481489511
6	2535.891113	2537.208496	2172.30127	2440.352295	2440.352295	28220804648
7	2440.393555	2498.50708	2359.384766	2455.935059	2455.935059	16179776932
8	2455.579102	2705.78418	2417.683105	2468.030273	2468.030273	21229909340
9	2467.188477	2510.45166	2328.452148	2423.001221	2423.001221	16126522783
10	2421.646973	2548.778564	2367.83374	2547.092041	2547.092041	14666227351
11	2546.590576	2627.609375	2526.989746	2597.084717	2597.084717	11172062661
12	2598.564941	2631.408447	2550.460938	2603.466553	2603.466553	9501221177
13	2603.263428	2697.735352	2489.072266	2688.278809	2688.278809	13778234614
14	2687.898926	2802.31543	2682.621826	2792.117188	2792.117188	13194846235
15	2791.958984	2802.212158	2630.120361	2682.854004	2682.854004	13876301217
16	2682.226074	2712.482666	2587.783447	2679.162598	2679.162598	12755505065
17	2681.057617	2983.586914	2675.443848	2983.586914	2983.586914	18987223729
18	2984.446045	3054.130127	2966.781006	3014.648193	3014.648193	13102093957
19	3014.959717	3061.26123	2965.429932	3057.476074	3057.476074	9466018022

GRAPHS PLOTTED BETWEEN FEATURE AND TARGET VARIABLE:

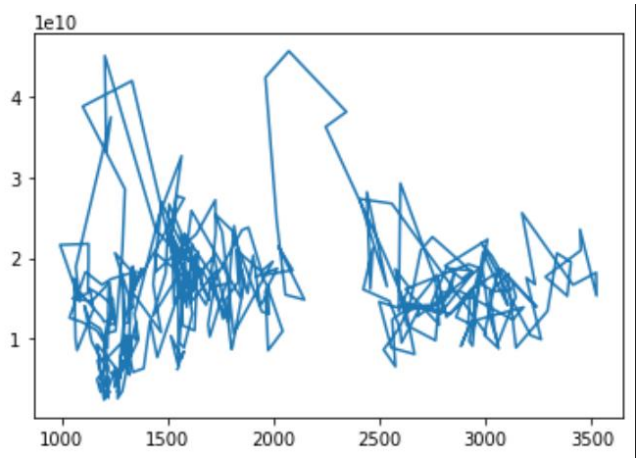
OPEN VS HIGH



LOW VS CLOSE



ADJ_CLOSE VS VOLUME



4. METHODOLOGY

4.1 PROCEDURE TO SOLVE THE GIVEN PROBLEM

In this project Ethereum price prediction we use four approaches:

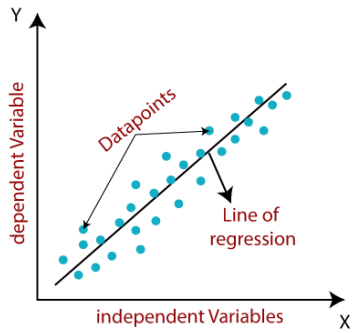
- Linear regression
- K-Nearest Neighbour
- Support Vector Machine
- Decision Tree

Linear Regression

Linear regression is a supervised machine learning method that is used by the Train Using AutoML tool and finds a linear equation that best describes the correlation of the explanatory variables with the dependent variable. This is achieved by fitting a line to the data using least squares. The line tries to minimize the sum of the squares of the residuals. The residual is the distance between the line and the actual value of the explanatory variable. Finding the line of best fit is an iterative process.

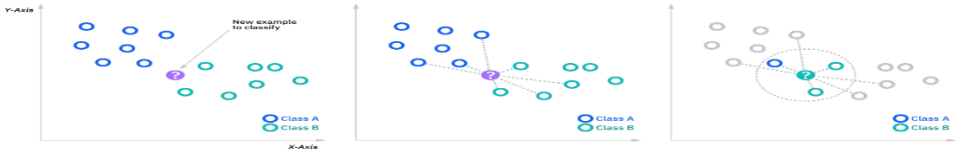
Advantages of linear regression algorithm:

- Linear regression performs exceptionally well for linearly separable data
- Easier to implement, interpret and efficient to train
- It handles overfitting pretty well using dimensionally reduction techniques, regularization, and cross-validation
- One more advantage is the extrapolation beyond a specific data set



K-Nearest Neighbour

The k-nearest neighbors algorithm, also known as KNN or k-NN, is a non-parametric, supervised learning classifier, which uses proximity to make classifications or predictions about the grouping of an individual data point. While it can be used for either regression or classification problems, it is typically used as a classification algorithm, working off the assumption that similar points can be found near one another.



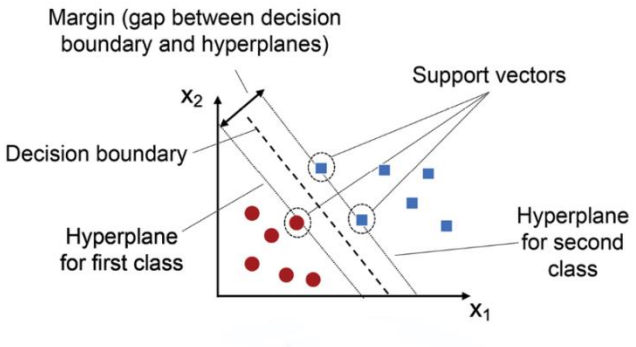
KNN Formula:

$$d(x,y) = \sqrt{\sum_{i=1}^{n} (y_i - x_i)^2}$$

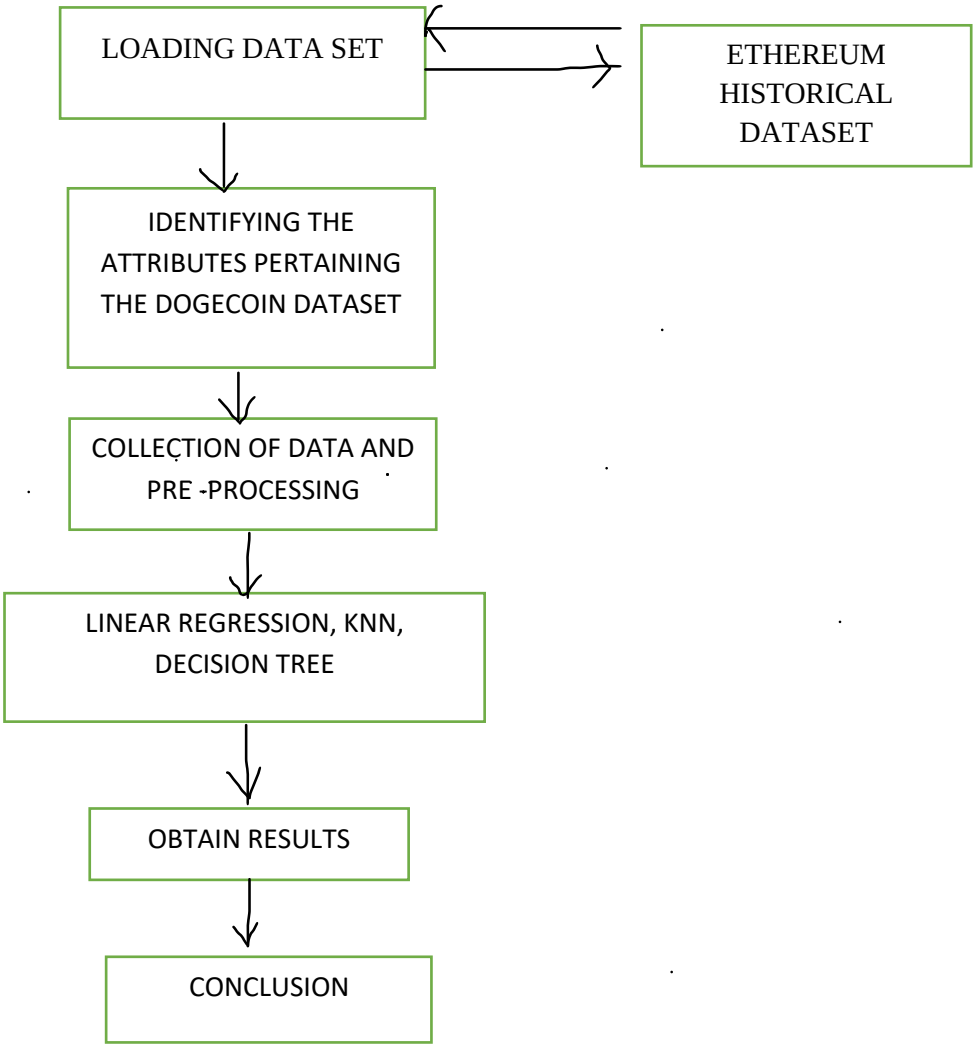
Support Vector Machine

Support Vector Machine or SVM is one of the most popular Supervised Learning algorithms, which is used for Classification as well as Regression problems. However, primarily, it is used for Classification problems in Machine Learning.

The goal of the SVM algorithm is to create the best line or decision boundary that can segregate n-dimensional space into classes so that we can easily put the new data point in the correct category in the future. This best decision boundary is called a hyperplane.



4.2 MODEL ARCHITECTURE



4.3 SOFTWARE DESCRIPTION

Software requirements:

Operating system: Windows

Platform: Google Collab

Programing language: Python

5. RESULTS

CODE

Dataset:

```
import pandas as pd
d=pd.read_csv("/content/drive/MyDrive/Ethereum_Dataset.csv")
print(d)
```

Output:

```
            Open         High          Low        Close    Adj_Close
0    3095.271729  3265.336914  3000.908203  3001.120117  3001.120117
1    3002.956787  3029.081055  2496.812988  2557.931641  2557.931641
2    2561.145264  2615.247314  2330.247314  2405.181152  2405.181152
3    2406.924316  2542.144775  2381.515137  2535.039063  2535.039063
4    2535.891113  2537.208496  2172.301270  2440.352295  2440.352295
..           ...          ...          ...          ...          ...
361  1552.519287  1594.039795  1529.573608  1576.833496  1576.833496
362  1577.107422  1594.000977  1553.335938  1567.846069  1567.846069
363  1567.698975  1602.106689  1509.422852  1515.506958  1515.506958
364  1515.249634  1557.970337  1514.380005  1552.556519  1552.556519
365  1550.773193  1559.552490  1545.327393  1545.741333  1545.741333

          Volume
0    10645922764
1    26796291874
2    27369692036
3    16481489511
4    28220804648
..           ...
361   8454485431
362   7599462786
363  10354880595
364   6432638856
365   6211551232

[366 rows x 6 columns]
```

Linear Regression:

```
from sklearn.linear_model import LinearRegression

from sklearn.metrics import
confusion_matrix,accuracy_score,r2_score,classification_report,mean_squared_
error

model=LinearRegression()

model.fit(x_train,y_train)

y_pred=model.predict(x_test)
```

Output:

```
[1.29795606e+10 1.03126207e+10 1.40868204e+10 2.59244427e+10
 1.36039946e+10 1.00096283e+10 1.29234393e+10 1.00939320e+10
 1.24575902e+10 2.93797008e+10 1.63024013e+10 1.71549651e+10
 1.12948154e+10 1.56232087e+10 1.16661963e+10 1.07511953e+10
 2.85511733e+10 1.37553749e+10 1.79700056e+10 1.06518591e+10
 2.03198830e+10 1.11396145e+10 1.42979493e+10 2.71326593e+10
 1.02016645e+10 9.80941018e+09 1.46959574e+10 9.46784242e+09
 2.28149654e+10 1.15335002e+10 1.17498761e+10 1.14498178e+10
 2.98152585e+10 1.32610775e+10 1.64936548e+10 1.61638836e+10
 1.39579309e+10 2.09930347e+10 9.66941588e+09 1.19058940e+10
 1.46114825e+10 1.01748014e+10 1.63794648e+10 1.06228466e+10
 1.97533740e+10 1.24947075e+10 1.30901876e+10 1.35628571e+10
 1.21126007e+10 8.29614561e+09 1.28098041e+10 1.33242773e+10
 1.30465432e+10 1.13783650e+10 1.07664852e+10 1.87867239e+10
 1.21748838e+10 1.14103263e+10 1.12142345e+10 2.17842509e+10
 1.03476472e+10 1.08535199e+10 1.92906046e+10 1.31498309e+10
 1.45633748e+10 9.33838806e+09 1.05892675e+10 1.29040098e+10
 1.21368640e+10 1.51591586e+10 1.14502391e+10 9.50602660e+09
 1.45102667e+10 1.45440536e+10]
```

K-Nearest Neighbour:

```
from sklearn.neighbors import KNeighborsRegressor
model=KNeighborsRegressor()
model.fit(x_train,y_train)
y_pred=model.predict(x_test)
print(y_pred)
from sklearn.metrics import mean_squared_error
print(mean_squared_error(y_test,y_pred))
```

Output:

```
[1.22833407e+10 1.26673718e+10 1.49429881e+10 1.54902880e+10
 1.38284031e+10 4.38490145e+09 1.58937769e+10 5.04153299e+09
 1.73970542e+10 1.74068861e+10 1.45495977e+10 1.82095116e+10
 9.19587186e+09 2.03626007e+10 4.14255156e+09 1.86713173e+10
 1.46977515e+10 1.38325160e+10 1.89133986e+10 1.38954922e+10
 2.14329781e+10 1.68538595e+10 1.28672225e+10 1.21875303e+10
 4.38490145e+09 8.32735586e+09 1.96325995e+10 4.51567529e+09
 1.66164456e+10 1.60791187e+10 1.50095414e+10 1.48039111e+10
 1.65735180e+10 1.38325160e+10 1.11897801e+10 2.02664625e+10
 1.36673360e+10 2.14503664e+10 7.44696884e+09 1.57689057e+10
 1.53326780e+10 4.33723572e+09 1.71048551e+10 1.33987999e+10
 1.44837241e+10 1.50852695e+10 1.68522273e+10 1.59562575e+10
 1.73970542e+10 1.57689057e+10 1.46775786e+10 1.64406071e+10
 1.78622249e+10 1.53359775e+10 1.32881596e+10 1.30810574e+10
 1.36469415e+10 1.29753444e+10 1.35177538e+10 1.68215966e+10
 4.37701995e+09 8.50334321e+09 2.02664625e+10 1.31239195e+10
 1.20262423e+10 4.93270734e+09 1.38325160e+10 1.56529009e+10
 1.95614671e+10 9.06936519e+09 1.83744726e+10 3.77784164e+09
 1.56529009e+10 1.03709722e+10]
2.5395746204151992e+19
```

Support Vector Machine:

```
from sklearn.svm import SVR
model=SVR()
model.fit(x_train,y_train)
y_pred=model.predict(x_test)
print(y_pred)
from sklearn.metrics import mean_squared_error
print(mean_squared_error(y_test,y_pred))
```

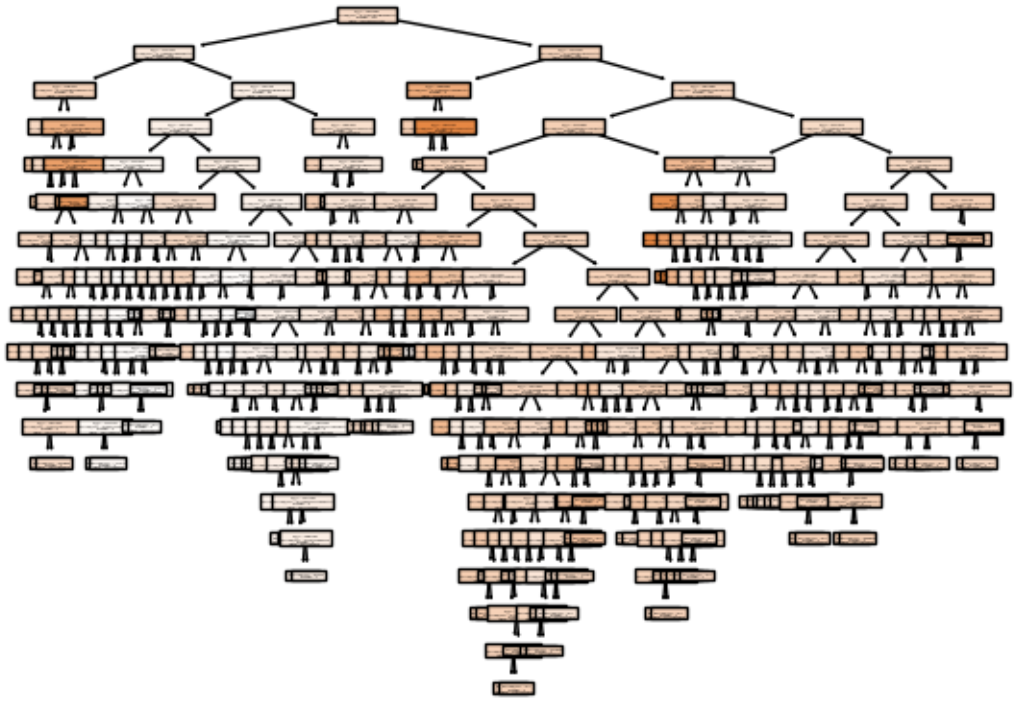
Output:

```
[1.4832442e+10 1.4832442e+10 1.4832442e+10 1.4832442e+10 1.4832442e+10
 1.4832442e+10 1.4832442e+10 1.4832442e+10 1.4832442e+10 1.4832442e+10
 1.4832442e+10 1.4832442e+10 1.4832442e+10 1.4832442e+10 1.4832442e+10
 1.4832442e+10 1.4832442e+10 1.4832442e+10 1.4832442e+10 1.4832442e+10
 1.4832442e+10 1.4832442e+10 1.4832442e+10 1.4832442e+10 1.4832442e+10
 1.4832442e+10 1.4832442e+10 1.4832442e+10 1.4832442e+10 1.4832442e+10
 1.4832442e+10 1.4832442e+10 1.4832442e+10 1.4832442e+10 1.4832442e+10
 1.4832442e+10 1.4832442e+10 1.4832442e+10 1.4832442e+10 1.4832442e+10
 1.4832442e+10 1.4832442e+10 1.4832442e+10 1.4832442e+10 1.4832442e+10
 1.4832442e+10 1.4832442e+10 1.4832442e+10 1.4832442e+10 1.4832442e+10
 1.4832442e+10 1.4832442e+10 1.4832442e+10 1.4832442e+10 1.4832442e+10
 1.4832442e+10 1.4832442e+10 1.4832442e+10 1.4832442e+10 1.4832442e+10
 1.4832442e+10 1.4832442e+10 1.4832442e+10 1.4832442e+10 1.4832442e+10
 1.4832442e+10 1.4832442e+10 1.4832442e+10 1.4832442e+10 1.4832442e+10
 1.4832442e+10 1.4832442e+10 1.4832442e+10 1.4832442e+10]
3.627272436390147e+19
```

Decision Tree:

```
from sklearn.tree import DecisionTreeRegressor
model=DecisionTreeRegressor()
model.fit(x_train,y_train)
y_pred=model.predict(x_test)
print(y_pred)
from sklearn.metrics import mean_squared_error
print(mean_squared_error(y_test,y_pred))
from sklearn import tree
tree.plot_tree(model,filled=True)
```

Output:



6. CONCLUSION AND FUTURE SCOPE

- Predicting the future price and trading activity of Ethereum can be challenging due to the highly volatile nature of the cryptocurrency market.
- Analyzing the adjusted close and volume graph plots can provide valuable insights into the historical trend of Ethereum's price and trading activity, which can be useful in making informed investment decisions.
- However, it's important to remember that any predictions based on historical data should be taken with caution and should not be relied upon solely to make investment decisions.
- As with any investment, it's important to do your own research, consult with a financial advisor, and carefully consider your personal financial goals and risk tolerance before investing in Ethereum or any other cryptocurrency.

7.REFERENCES

a. Tandon, R., Khurana, H., Kumar, S., & Singh, S. (2021). Ethereum Price Prediction with Sentiment Analysis for Smart Trading. Journal of Ambient Intelligence and Humanized Computing, 12(5), 4895-4908.

b. Kovanen, V., Kivelä, M., Saramäki, J., & Kaski, K. (2018). Bitcoin and Ethereum price forecast using historical trading volumes and network parameters. Journal of Computational Science, 28, 343-348.

c. Ge, C., & Lai, C. (2018). Forecasting Cryptocurrency Prices with Machine Learning. IEEE Transactions on Emerging Topics in Computational Intelligence, 3(3), 246-258.