# **Ethereum Price Movement Prediction**

This Python project predicts the direction of Ethereum (ETH-USD) price movements (up or down) for the next month using historical price data and technical indicators, analyzed with a machine learning model. It leverages libraries like pandas, numpy, yfinance, matplotlib, ta, and scikit-learn to fetch data, calculate indicators, train a RandomForestClassifier, and visualize results.

## **Purpose**

The goal is to forecast whether Ethereum’s price will increase (1) or decrease (0) in the upcoming month based on monthly historical data and technical indicators such as Moving Averages, RSI, Bollinger Bands, and MACD. The project also provides visualizations and an analysis of key indicators influencing the prediction.

## **Prerequisites**

To run this project, ensure you have the following installed:

* **Python 3.x**: The project requires a Python environment (3.6 or higher recommended).
* **Required Libraries**:
  + pandas: For data manipulation and analysis.
  + numpy: For numerical computations.
  + yfinance: To fetch Ethereum price data from Yahoo Finance.
  + matplotlib: For plotting historical data and indicators.
  + ta: For calculating technical analysis indicators.
  + scikit-learn: For the RandomForestClassifier and accuracy metrics.

## **Installation**

Follow these steps to set up the project:

1. **Save the Script**: Save your Python script (e.g., as eth\_price\_prediction.py) in a local directory.
2. **Install Dependencies**: Open a terminal and run the following command to install all required libraries:  
    bash  
   CollapseWrapCopy  
   pip install pandas numpy yfinance matplotlib scikit-learn ta
3. **Verify Installation**: Ensure all libraries are installed by running a Python interpreter and importing them:  
    python  
   CollapseWrapCopy  
   import pandas, numpy, yfinance, matplotlib, ta, sklearn

## **Usage**

To execute the project and generate predictions:

1. **Ensure an Internet Connection**: The script fetches live data from Yahoo Finance.
2. **Run the Script**: From the terminal, navigate to the directory containing your script and run:  
    bash  
   CollapseWrapCopy  
   python eth\_price\_prediction.py

### **What to Expect**

When you run the script, it will:

* Download historical Ethereum price data (ETH-USD) from Yahoo Finance, starting from January 1, 2020, to the present.
* Resample the data to monthly intervals.
* Calculate technical indicators (e.g., SMA, EMA, RSI, Bollinger Bands, MACD).
* Train a RandomForestClassifier to predict the next month’s price direction.
* Display plots of historical price, RSI, and Bollinger Bands.
* Output the prediction for the next month and analyze key indicators.

### **Sample Output**

text

CollapseWrapCopy

Fetching Ethereum data...

Calculating technical indicators...

Training model and predicting next month...

Prediction for Next Month (e.g., March 2025): Uptrend

Model Accuracy: 0.94

*Note: Actual output depends on the full script implementation and the date of execution.*

## **Project Structure**

The script begins with importing libraries and is likely structured into several key steps (inferred from the provided snippet and typical ML workflows):

### **1. Import Libraries and Setup**

python

CollapseWrapCopy

import pandas as pd

import numpy as np

import yfinance as yf

import matplotlib.pyplot as plt

from ta import add\_all\_ta\_features

from ta.trend import SMAIndicator, EMAIndicator, MACD

from ta.momentum import RSIIndicator

from ta.volatility import BollingerBands

from sklearn.ensemble import RandomForestClassifier

from sklearn.metrics import accuracy\_score

from datetime import datetime

This section sets up the environment by importing tools for data handling, fetching, technical analysis, machine learning, and visualization.

### **2. Fetch Ethereum Data**

* Uses yfinance to download daily ETH-USD data, likely resampled to monthly intervals (e.g., first open, max high, min low, last close).

### **3. Calculate Technical Indicators**

* Employs the ta library to add indicators such as:
  + **SMA**: Simple Moving Average (e.g., 20-month).
  + **EMA**: Exponential Moving Average (e.g., 20-month).
  + **RSI**: Relative Strength Index (e.g., 14-month).
  + **Bollinger Bands**: 20-month window with 2 standard deviations.
  + **MACD**: Moving Average Convergence Divergence.
* Likely uses add\_all\_ta\_features for a broad set of indicators, with specific ones extracted via SMAIndicator, EMAIndicator, etc.

### **4. Preprocess Data**

* Prepares the data for machine learning by creating a target variable (e.g., 1 for price increase, 0 for decrease) and selecting features (indicators).

### **5. Train Model and Predict**

* Trains a RandomForestClassifier on historical data (excluding the last row) to predict the next month’s price movement.
* Evaluates performance using accuracy\_score (on training data, ideally).

### **6. Visualize Results**

* Uses matplotlib.pyplot to plot historical prices and indicators (e.g., RSI, Bollinger Bands).

*Note: The exact structure beyond Step 1 depends on your full implementation, but this outlines a typical flow based on the libraries.*

## **Notes and Limitations**

* **Assumption on "tc rth"**: The README assumes "tc rth" refers to Ethereum (ETH-USD), as inferred from yfinance usage. If this is incorrect, update the README accordingly.
* **Internet Dependency**: Requires a live connection to fetch data from Yahoo Finance.
* **Prediction Scope**: Likely predicts only direction, not magnitude, unless otherwise implemented.
* **Model Evaluation**: If accuracy is computed only on training data, it may overestimate performance. Consider adding a train-test split or cross-validation.
* **Indicator Selection**: While add\_all\_ta\_features generates many indicators, only a subset (e.g., SMA, EMA, RSI, BB, MACD) may be used explicitly—clarify in your script.
* **Potential Improvements**:
  + Tune RandomForest hyperparameters (e.g., number of trees, max depth).
  + Add alternative models (e.g., SVM, LSTM).
  + Incorporate external data (e.g., market sentiment).

## **Contributing**

Feel free to fork this project, enhance the model, or add new features. Submit pull requests or raise issues for suggestions!

## **License**

This project is open-source and available under the [MIT License](https://opensource.org/licenses/MIT) (or specify your preferred license).