

BITCOIN -ANALYSIS

“Bitcoin” is one of the most widely traded cryptocurrencies, and its price movements are highly volatile.

Predicting Bitcoin’s price using historical data can provide valuable insights for traders and investors.

In this document , we will outline a comprehensive approach to building a bitcoin price prediction model in “tableau”

Step-1 : Loading Bitcoin data into Tableau

To create a forecasting model and visualize insights and trends , the first step is load the data into Tableau.

Tableau Public - Bitcoin

FileDataWindowHelp

Tableau Public Desktop

Upgrade

Bitcoin

Microsoft Excel

ConnectionsAdd

Sheets

Bitcoin

New Union

New Table Extension

Bitcoin (Bitcoin)

Filters0Add

Bitcoin

13 fields 3836 rows

100rows

Name

Bitcoin

Fields

Type	Field Name	Phys...	Rem...
#	Open	Bitcoin	Open
#	High	Bitcoin	High
#	Low	Bitcoin	Low
#	Close	Bitcoin	Close
#	Volume	Bitcoin	Volume
#	Divide...	Bitcoin	Divide...
#	Stock ...	Bitcoin	Stock ...

#	Bitcoin F1	Bitcoin Date	Bitcoin Open	Bitcoin High	Bitcoin Low	Bitcoin Close	Bitcoin Volume	Bitcoin Dividends
0	9/17/2014	465.864	468.174	452.422	457.334	21,056,800	0	
1	9/18/2014	456.860	456.860	413.104	424.440	34,483,200	0	
2	9/19/2014	424.103	427.835	384.532	394.796	37,919,700	0	
3	9/20/2014	394.673	423.296	389.883	408.904	36,863,600	0	
4	9/21/2014	408.085	412.426	393.181	398.821	26,580,100	0	
5	9/22/2014	399.100	406.916	397.130	402.152	24,127,600	0	
6	9/23/2014	402.092	441.557	396.197	435.791	45,099,500	0	
7	9/24/2014	435.751	436.112	421.132	423.205	30,627,700	0	
8	9/25/2014	423.156	423.520	409.468	411.574	26,814,400	0	
9	9/26/2014	411.429	414.938	400.009	404.425	21,460,800	0	
10	9/27/2014	403.556	406.623	397.372	399.520	15,029,300	0	

Data Source

open as per year

high as per year

low per year

close per year

OHLC per year

MA of close trends

gain or loss

large price movement

trade volume

gain/loss price move

op

fields

help

Bitcoin-dataset

The dataset contain following requirements:

Date: The timestamp of Bitcoin prices.

Open : Open price of Bitcoin for each day.

Close: Close price of Bitcoin for each day.

High: Highest price of Bitcoin for each day.

Low: Lowest price of Bitcoin for each day.

Volume: the trading Volume of the Bitcoin for each day.

Dividends: Dividends of Bitcoin for each day.

Step-2: Creating a Time Series Chart

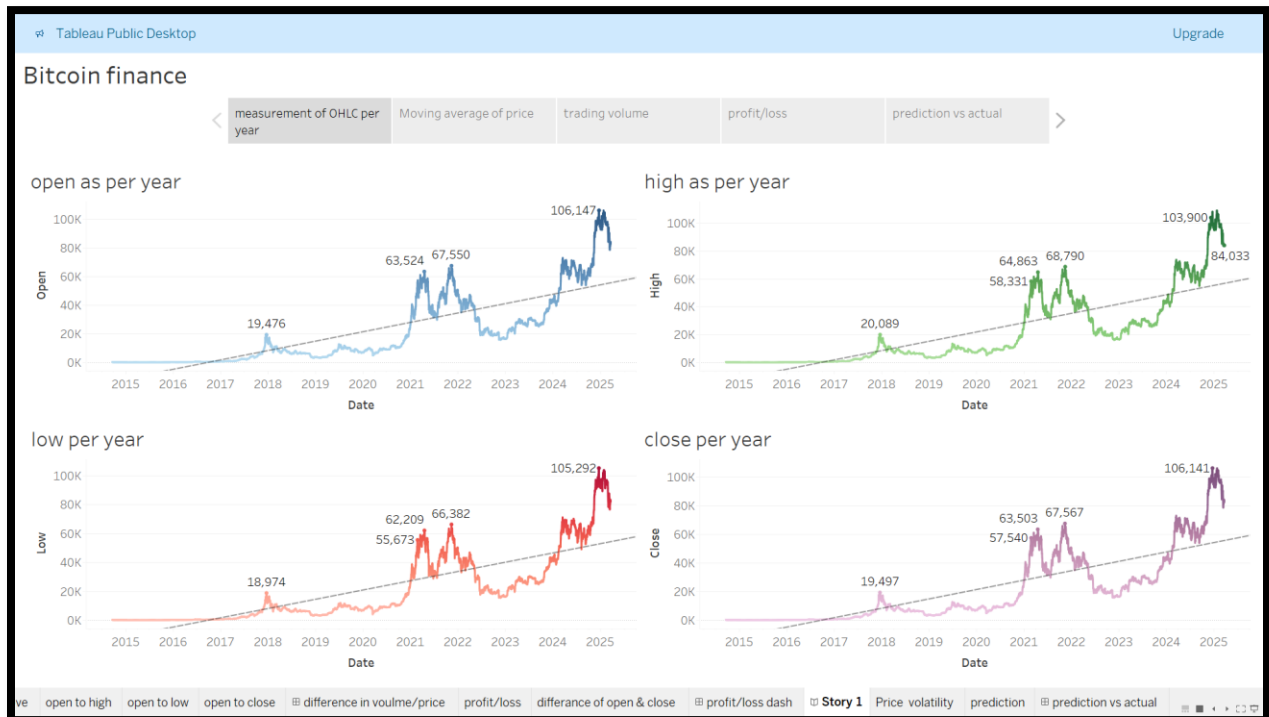
Drag Date to columns.

Drag Open , Close, Low, and Close price of Bitcoin to rows seperatly.

Choose the Line-Chart.

Add trend line.

Drag Open , Close, Low and High price to Color.



Open per year

High per year

Low per year

Close per year

Creating all four charts and then make a dashboard from it.

Step-3: Moving Average and large price movement for Gain & Loss

1) Moving-Average:

Create Calculated- field

`WINDOW_AVG(SUM([Close]), -6, 0)`

Drag this field to rows.

Drag date field to column.

Choose dual axis chart.

Add trend line.

Drag calculated-field to colors.

2) Gain-Loss chart:

Drag date field to column.

Create Gain and loss calculated field .

Drag gain and loss field to column.

Drag sum of High , low and open field to row.

Choose the bar chart.

3) Large moving price:

Drag date field to column.

Create a calculated-field:

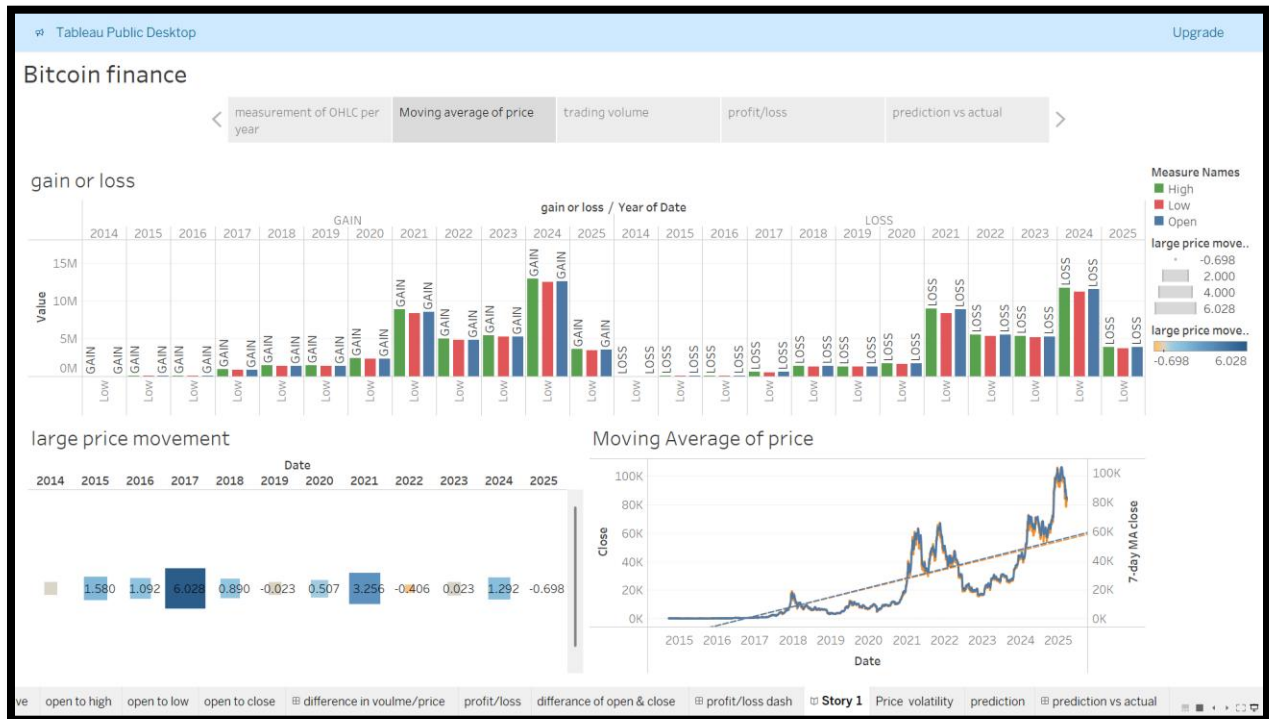
`(SUM([Close]) - LOOKUP(SUM([Close]), -1))/ LOOKUP(SUM([Close]), -1)`

Drag this field to column.

Drag this field to color.

Drag this field to label.

Drag this field to detail.



Step-4: Volume and difference chart

1): Open to High chart:

Drag date field to column.

Create measure name field by Open and High field.

Drag measure name field to color.

Choose bar chart.

2): Open to low chart:

Drag date field to column.

Create measure name field by Open and Low field.

Drag measure name field to color.

Choose bar chart.

3): Open to Close chart:

Drag date field to column.

Create measure name field by Open and Close field.

Drag measure name field to color.

Choose bar chart.

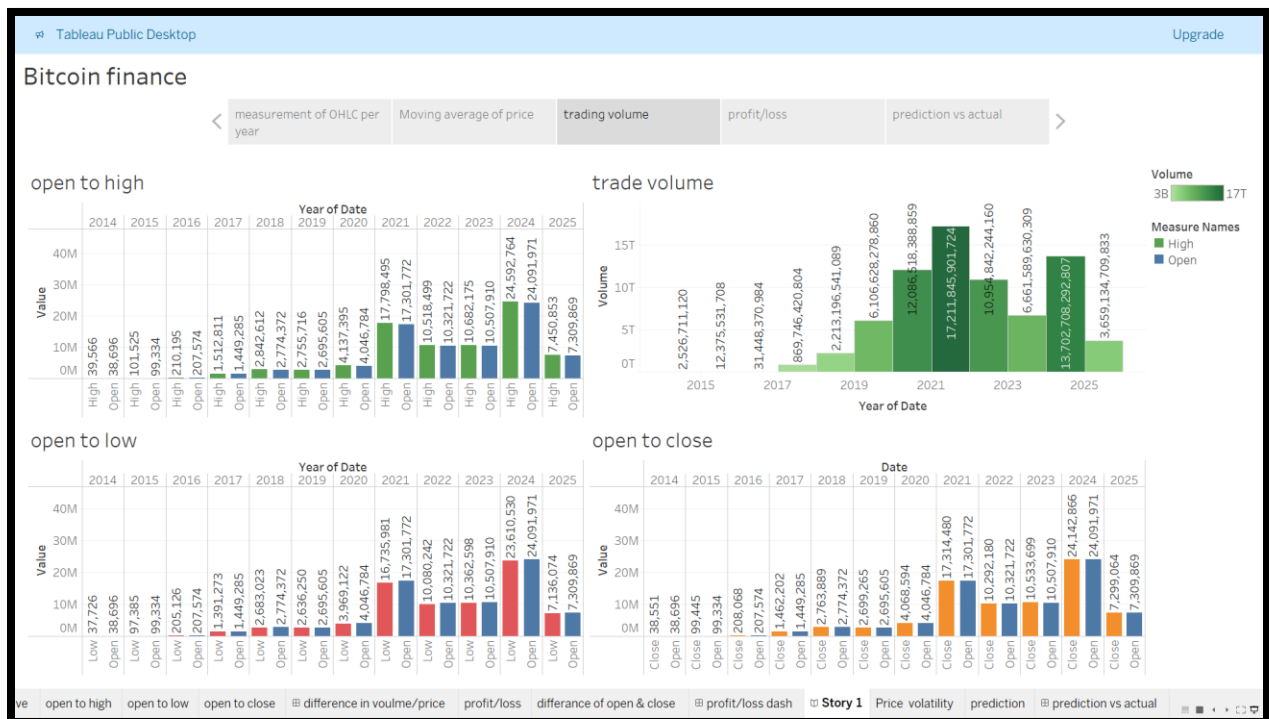
4): Volume Chart:

Drag date field to column.

Drag volume field to row.

Drag volume field to color.

Choose bar chart.



Step-5 : Profit-loss charts

1): profit/loss chart

Create profit and loss calculated field.

Drag those field to row.

Drag date field to column.

Select forecast .

Drag High and open to color.

2): Difference chart of Open and Close

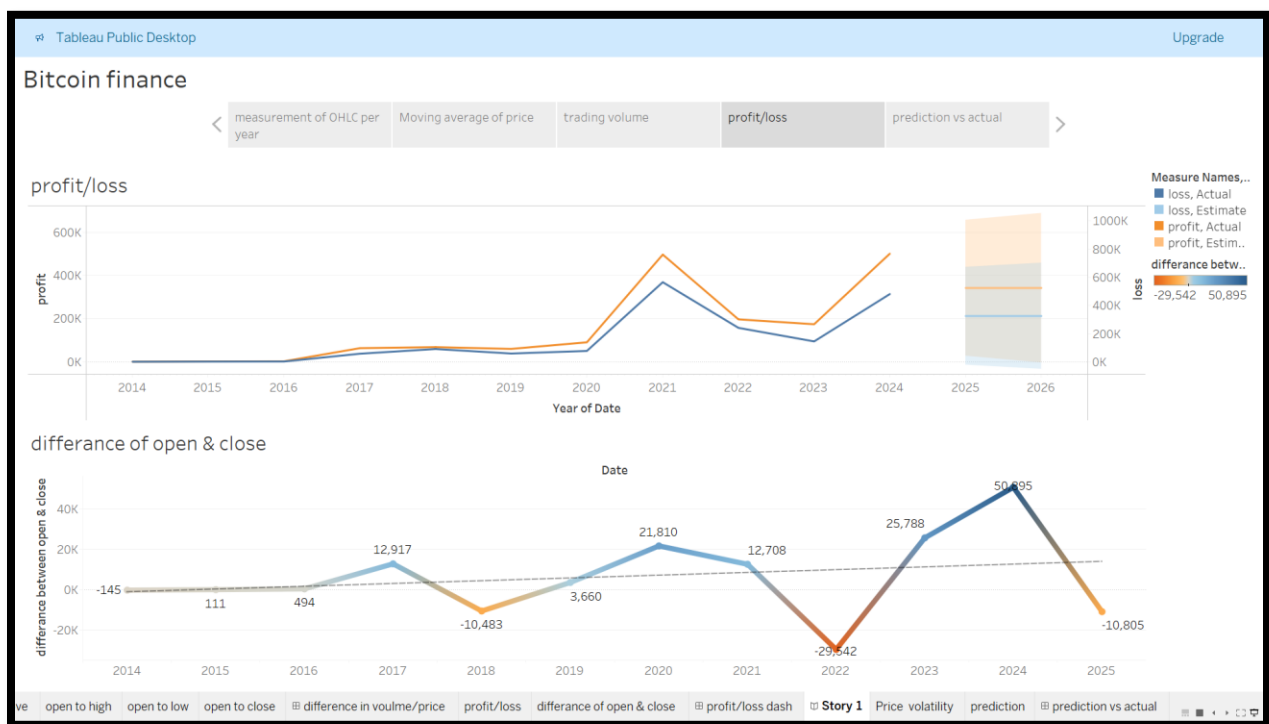
Drag date field to column.

Create a calculated field .

Drag calculated field to row.

Drag calculated field to label.

Drag calculated field to color.



Step-6: Prediction and price Volatility chart

1) Prediction Chart

Drag date field to column.

Drag close field to row.

Add trend lines.

Select forecast .

Drag forecast to color.

2) Volatility chart

Drag date field to column.

Drag close field to row.

Create middle band calculation field:

$\text{WINDOW_AVG}(\text{SUM}(\text{Close}), -19, 0)$

Create upper band calculation field:

$[\text{WINDOW_AVG}(\text{SUM}(\text{Close}), -19, 0) + (2 * \text{WINDOW_STDEV}(\text{SUM}([\text{Close}]), -19, 0)]$

Create lower band calculation field:

$[\text{WINDOW_AVG}(\text{SUM}(\text{Close}), -19, 0) - (2 * \text{WINDOW_STDEV}(\text{SUM}([\text{Close}]), -19, 0)]$

Drag middle band, upper band, and lower band to row .

Then make dual axis chart.

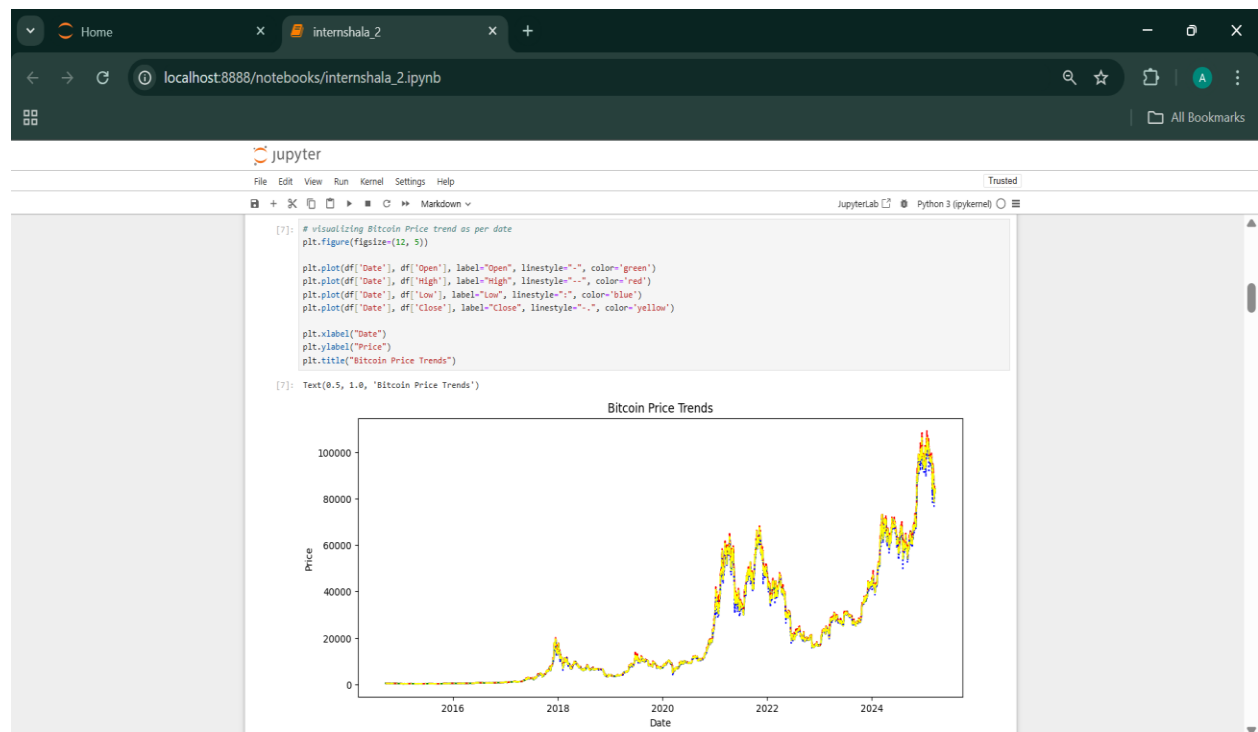
And then select synchronise axis.

And then drag measure name field to color.



Adjusting confidence interval to 95%.

Now enhancing those data analysis with machine-learning concept:



Bitcoin price trend chart

Following are key points:

the highest high bitcoin rate is 109114.8828125 at 20 January 2025

the lowest high bitcoin rate is 211.73100280761 at 17 january 2015

the highest open bitcoin rate is 106147.296875 at 18 December 2024

the lowest open bitcoin rate is 176.897003173828 at 15 january 2015

the highest low bitcoin rate is 105291.734375 at 17 december 2024

the lowest low bitcoin rate is 171.509994506835 at 14 january 2015

the highest close bitcoin rate is 106146.265625 at 21 january 2025

the lowest close bitcoin rate is 178.102996826171 at 14 january 2015



Some prediction dates:

```
make some predictions

[27]: df_period['Timestamp'] = df_period['Date'].astype('int64') // 10**9 # Convert Date to UNIX timestamp

X = df_period['Timestamp'].values.reshape(-1, 1)
y = df_period['Close']

# Train the model
model = LinearRegression()
model.fit(X, y)

future_dates = pd.date_range(start='2026-01-01', periods=5, freq='Y')
future_timestamps = future_dates.astype('int64') // 10**9

predictions = model.predict(future_timestamps.values.reshape(-1, 1))

for date, price in zip(future_dates, predictions):
    print(f"{date.date()} : {price:.2f}")

2026-12-31 : 65230.63
2027-12-31 : 71633.64
2028-12-31 : 78054.19
2029-12-31 : 84457.20
2030-12-31 : 90860.20

C:\Users\hvp\AppData\Local\Temp\ipykernel_10280\3499983980.py:1: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row indexer,col indexer] = value instead
```

2026-12-31 : 65230.63

2027-12-31 : 71633.64

2028-12-31 : 78054.19

2029-12-31 : 84457.20

2030-12-31 : 90860.20



end

