Bitcoin Price Prediction

Exploratory deep learning project using Bitcoin and altcoin time-series data

Motivation

While exploring my interests in machine learning, I encountered deep learning libraries such as TensorFlow and PyTorch. I wanted a hands-on way to learn how these tools work in practice, particularly on noisy, real world data. I chose Cryptocurrency data, as it is notoriously volatile and unpredictable, making it a challenging environment for experimentation.

Goal and Questions

The primary goal of this project was exploration. Rather than optimizing performance, I used cryptocurrency data as a playground to better understand time-series modeling and deep learning behavior.

A few questions I aimed to answer are:

1. How can I design a data pipeline that updates cleanly with new, recent data?
2. What failure modes do time-series models encounter when applied to highly noisy data?
3. How do deep learning models such as CNNs and LSTMs behave with this data?

Data + Pipeline

Data Ingestion

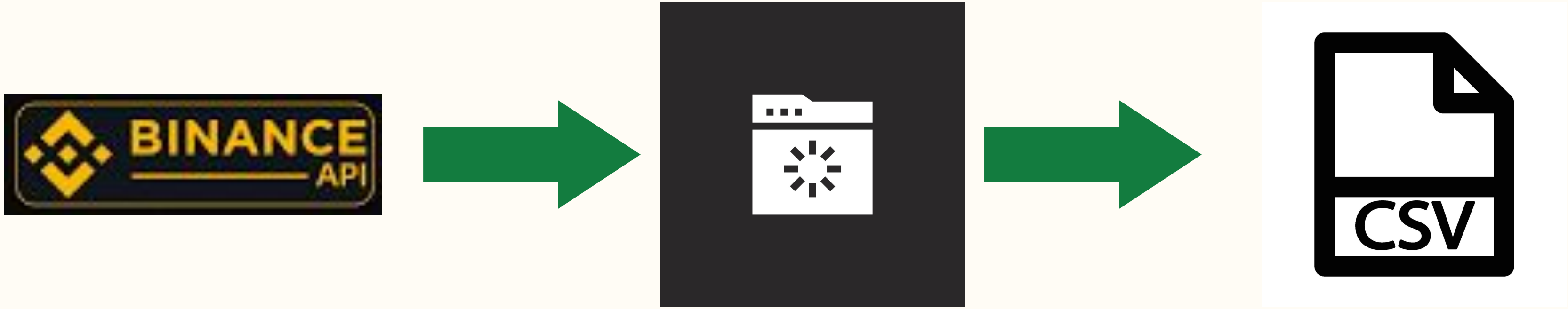
The pipeline begins by fetching hourly cryptocurrency market data from public APIs. Data is retrieved in a structured format and aligned to consistent timestamps to ensure reliability across updates. This step establishes a stable foundation for downstream processing.

Incremental Updating & Storage

Rather than refetching historical data, the pipeline is designed to append only newly available observations on each run. This incremental update strategy prevents duplication, keeps the dataset efficient, and reflects how live data systems operate in practice.

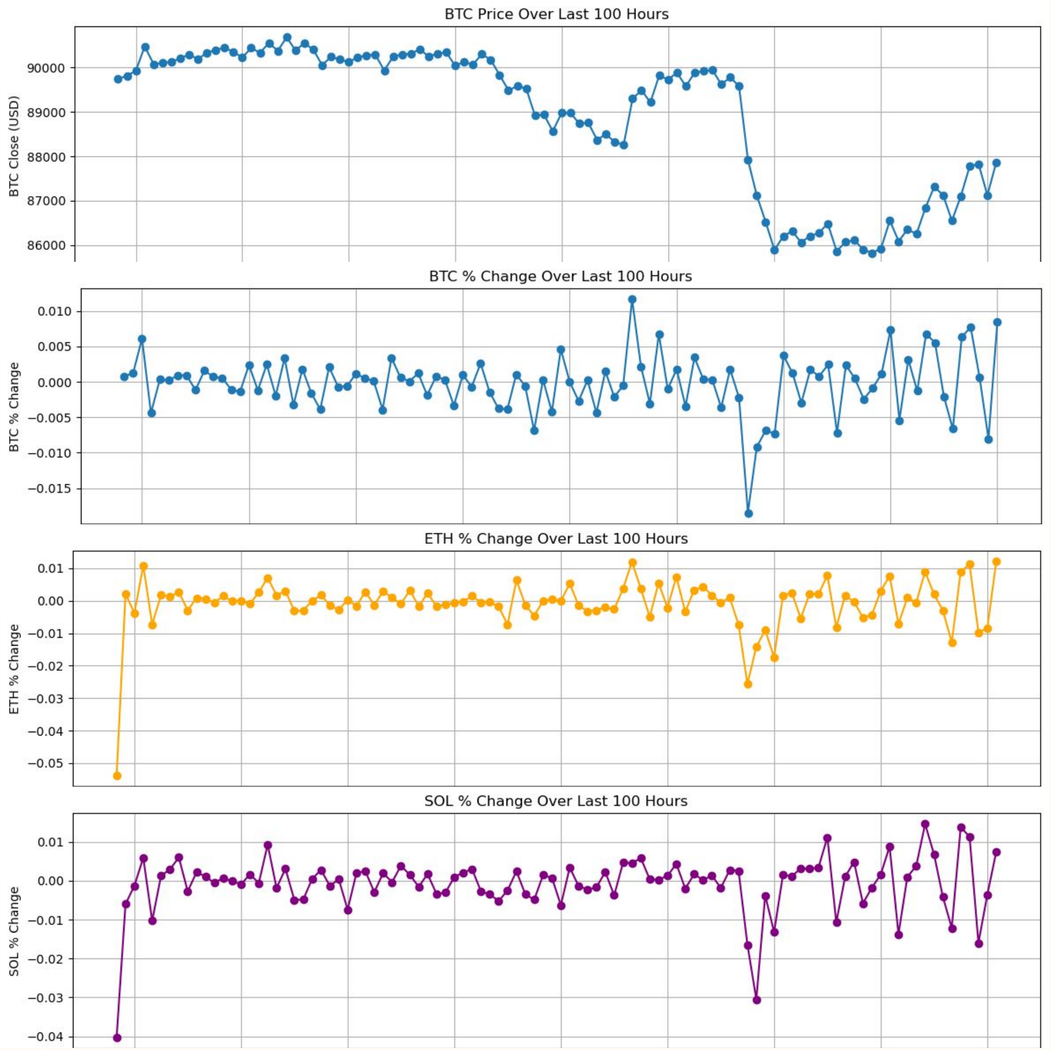
Feature Construction & Formatting

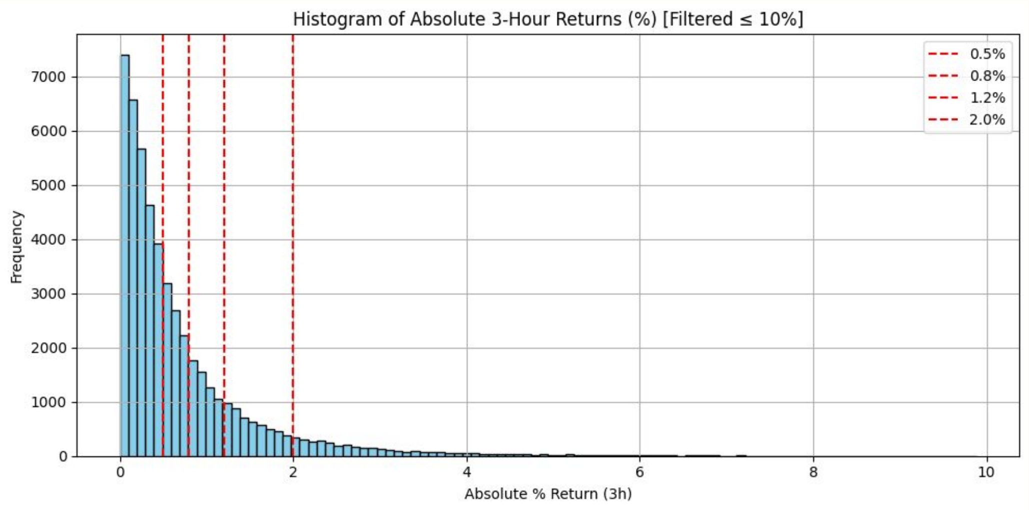
Raw price data is transformed into higher-level features such as returns, volatility, momentum indicators, and cross-asset signals. These features are standardized and organized into a consistent tabular format, ensuring the dataset is immediately usable for modeling and future experimentation.

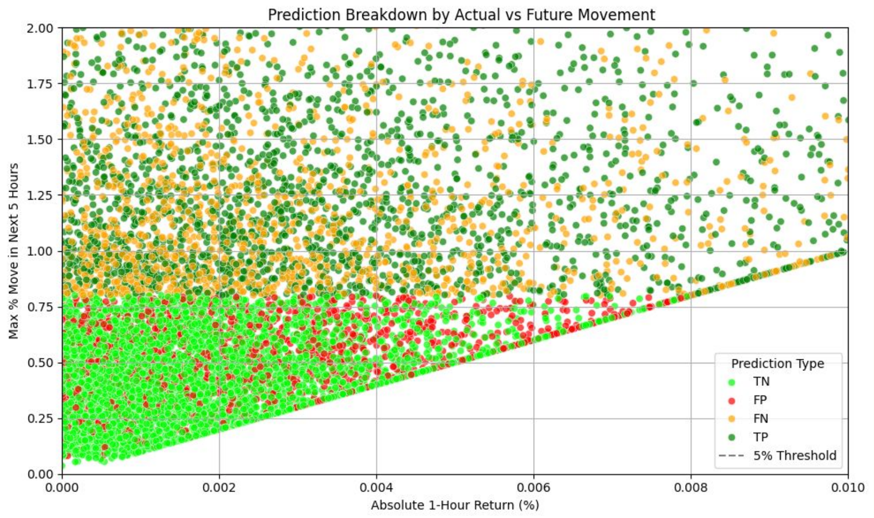


Approach

I initially attempted a simple time-series model to predict exact prices, but quickly realized this was not a practical goal for the project. I then shifted my focus from precise price prediction to extracting broader insights and trends from the data. As a first step, I moved away from raw prices and instead modeled percentage changes, which better captured relative movement and reduced scale related issues.







Modeling

I decided to move beyond single-hour predictions and instead focus on identifying whether a significant price movement occurred over the next few hours, using thresholds defined in the top graph. After labeling the data accordingly, I trained a CNN to predict these larger movements. The results were not statistically meaningful, as illustrated by the bottom visualization, which shows the model struggling to predict whether a change greater than 0.8 occurs in the next 5 hours.

Challenges

One major challenge was the extremely noisy nature of cryptocurrency data, which made it difficult for models to distinguish meaningful signal from random fluctuations. Even more complex models often defaulted to learning trivial patterns rather than genuine trends, especially my first LSTM, which seemed to track the data perfectly, until I realised it was just overfitting to the data. Additionally, evaluating model performance was nontrivial, as standard metrics did not always reflect whether the model had learned useful behavior or was simply exploiting short-term correlations.

What I Learned

This project showed me that in highly noisy data like cryptocurrency, model complexity alone does not lead to meaningful insight. How the data is represented and framed often matters more than the specific architecture used. I also learned that building a robust data pipeline is just as important as modeling itself, and that overfitting is not something that is guaranteed to show up in basic model evaluations.

What I'd Change

If I were to revisit this project, I would place more emphasis on shorter time frames rather than applying a single modeling approach across all data. I would also explore alternative prediction targets, such as directional movement or volatility changes, instead of magnitude alone. Finally, I would design more controlled experiments to better separate genuine learning from short-term correlations, for me this feels like the next step in becoming familiar with deep learning.

Why This Project?

This project deepened my interest in machine learning by emphasizing understanding over results. It strengthened my intuition for time-series modeling and highlighted the importance of careful experimentation. It was overall an engaging introduction to deep learning and data manipulation.