Machine Learning - Bitcoin Price Prediction

Presented by Trevor Huffstetler

link to video: **https://youtu.be/wayPjpiDuaU**

Project Overview

1

Problem and Challenges

Determine success of various models against low frequency and high frequency bitcoin datasets

2

Motivation and Related Approaches

Why did I choose my approach and what approaches are commonly used across the industry

3

Method

Data Collection

Feature Engineering

Model Creation

4

Results and observations

Comparison and performance across various models

5

Conclusion and future work

Problem Statement & Challenges

Problem Definition

I tackle determining which data frequencies and features impact various model performances

- data frequency and feature selection determine which models to use
- demand for price data with various details and more diverse feature dimensions

Challenges

- Crypto markets experience extreme volatility
- Heavily influenced by social sentiment
- Nature of Bitcoin is still elusive and widely debated

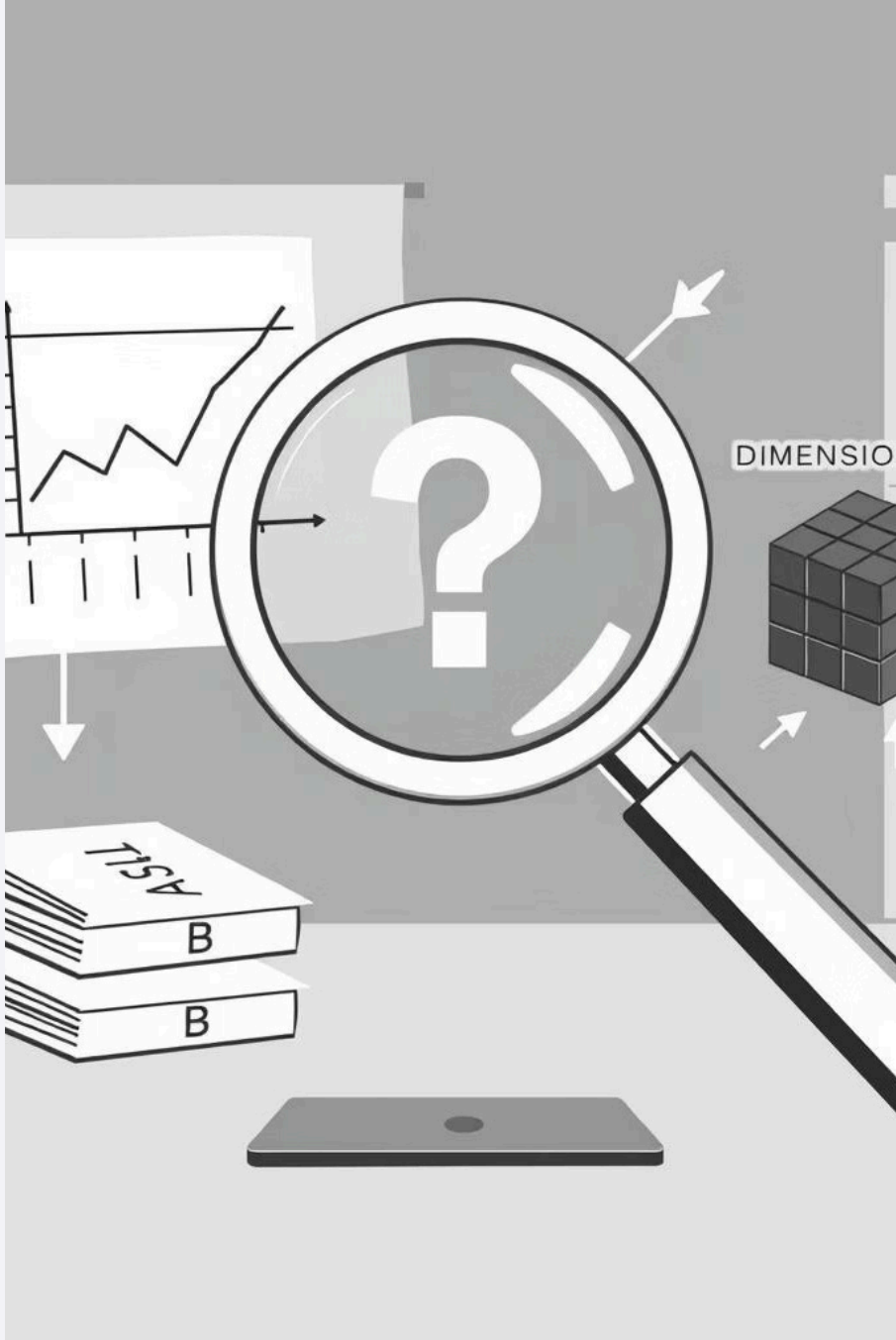
Motivation & Significance

The Problem

- Limited research focusing on the impact of data sample dimensions
- Models risk overfitting or missing crucial short-term patterns if sampling frequency and feature sets are overlooked

Solution

- Construct low-frequency(daily) and high frequency(5-minute) Bitcoin datasets to evaluate performance of various machine learning models and features



Existing Approaches & Limitations

Current Methods

- LTSM, RNN, Support Vector Machines, Random Forest
- Domain Expertise and empirical analysis to determine features
- Strong focus on model accuracy, often overlooking sample dimension choices

Weaknesses

- Subjectivity and uncertainty regarding feature selection
- Sample frequency and dimensionality are often not tuned for model fit
- Financial time series typically include a lot of noise

Existing Approaches & Limitations

Unrestricted Deep Learning with Social media indicators

- Ortu, M et al. (2022) applied deep learning models using crypto trading indicators and social media indicators scraped from Reddit and Github

Limitations

- Social media indicators significantly improve accuracy, but High-quality social media data often requires paid access
- Deep learning models are prone to overfitting to data
- Factors that affect Crypto markets can change quickly, making it difficult for deep learning to pick up on

Existing Approaches & Limitations

Analysis of Bitcoin Price Prediction Using Machine Learning

- Chen, J. (2023). engineered 47 variables and features across 8 categories and applied LTSM and Random Forest Regression for daily Bitcoin price
- Data split into two time periods for separate model training and evaluation

Limitations

- While the data is split temporally, the frequency(daily) remains the same,.
- Exclusively uses daily data and ignores **sample dimension engineering — potentially** missing short-term patterns.

Hypothesis: Engineer features based on granularity

The structure and frequency of Bitcoin price data significantly influence model performance.

- Complex Machine Learning models perform better with higher frequency data with lower dimensionality
- While low-frequency data (e.g., daily) often favors simpler statistical methods due to higher dimensionality and overfitting risks.
- Features must be customized to the data frequency and model type to optimize peformance

Bitcoin Price Prediction Architecture

Input: Two datasets

- High Frequency Bitcoin Price (every minute)
- Low Frequency Bitcoin Price (daily)

Output: binary variable prediction

-Upward movement: labeled as 1, which represents that the price will increase in the next time period

-Downward movement: labeled as 0, which represents the price will decrease in the next time period

Feature Extraction:

- High-dimensional features are applied to the Low Frequency Dataset
- Low-dimensional features are applied to the High Frequency Dataset

Methodology: Duplicated Approach

1

Data Preparation

- Loaded Bitcoin Price from Kaggle Dataset(include link/reference)
- Bitcoin Network data from **Blockchain.com**
- Bitcoin Google Trends

2

Feature Engineering

- Separated data into Daily(low frequency) and Minute (high frequency) datasets
  - created binary classification target variable
    - 5 minute price change target for minute dataset
    - Daily price change target for the daily frequency
- Created high dimensional features utilizing bitcoin price, trading, network data, and attention indicators and applied to the daily frequency dataset
- Created low dimensional features using only bitcoin price and trading indicators and applied to minute frequency dataset

3

Model Creation Optimization

Trained 4 supervised classifiers:

- Logistic Regressor
- LDA
- SVM
- RandomForest

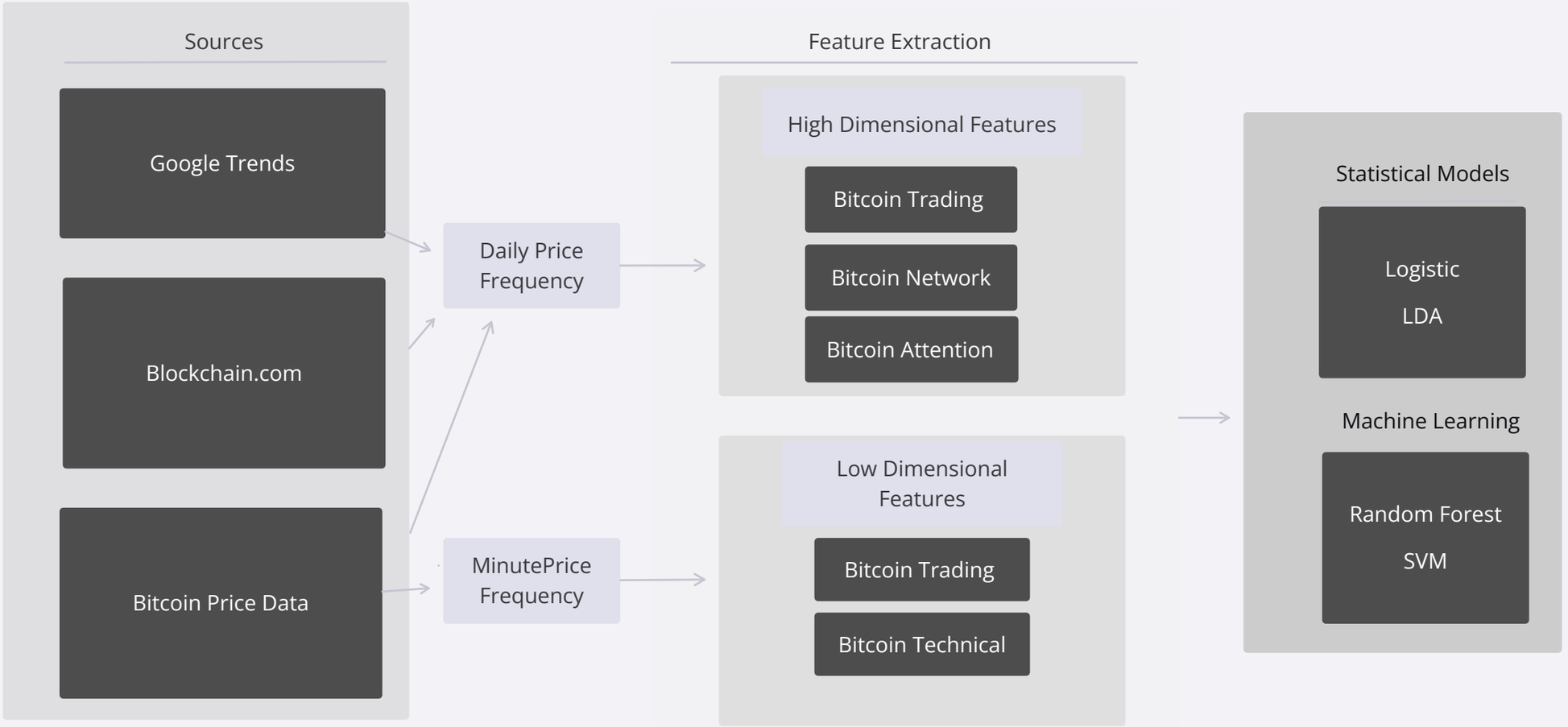
4

Evaluation

Confusion matrix measuring :

- Precision, Recall, F1-score
- Evaluated across both datasets (daily & minute)

Architecture for Bitcoin Price Prediction



Training Data

- To preserve the sequential nature of the financial data, the dataset was split using TimeSeriesSplit
- Each training set precedes the subsequent training set, mimicking a real world financial forecast

Feature Selection

1

Blockchain Network

1. Block Size
2. Transaction Difficulty
3. Hash Rate

2

Bitcoin Trading and Technical (StockStats Python Library)

1. Number of Transactions
2. Estimated Transaction Volume
3. Mempool Size
4. Mempool Transaction Count
5. Transaction Fees
6. Market Capitalization

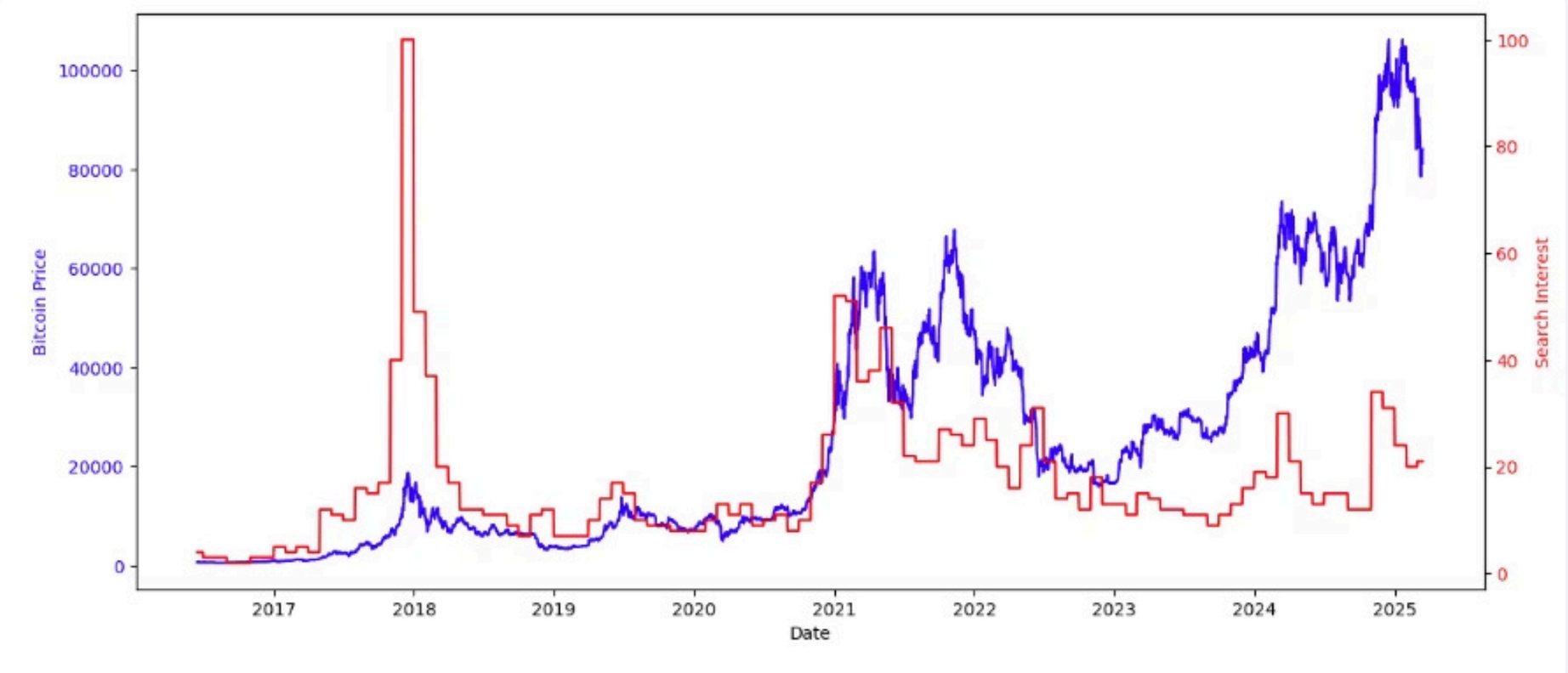
3

Bitcoin Attention

1. Google Trends
    a. historical google trend data is only available by month time period, so had to forward fill each months search interest values for each day in the month

Observations:

Plotting Google Trends against Bitcoin



This visualization suggest that there is strong connection between bitcoin price and Google search trends - a proxy for social sentiment. However, while the actual search interest values (0-100) don't perfectly align with the price, the changes in search interest over time tend to be more indicative of positive price movement than consistently high values alone.

*created in Python using Seaborn library

Parameter Settings

Used GridsearchCV to determine the best parameters to use for each model. The best performing parameters are listed below:

Logistic Regression Parameters

	C	penalty	solver
logreg_best_params	100	l2	liblinear

Linear Discriminant Analysis (LDA) Parameters

None

Support Vector Machine (SVM) Parameters

SVM Parameters

	C	gamma	kernel
SVM_best_params	10	scale	rbf

Random Forest (RF) Parameters

RF Parameters

	bootstrap	max_depth	max_features	min_samples_split	n_estimators
RF_best_params	False	5	sqrt	10	100

Results, Conclusion and Next Steps

My current status of the project is that I have obtained preliminary results and all models are performing around 50% accuracy.

After further fine tuning my models, my hypothesis is that I will confirm:

- LDA and Logistic models will perform better on lower frequency data
- SVM and Random Forrest will perform better on higher frequency data

Citations

Primary Research Paper:

Chen, Z., Li, C., & Sun, W. (2020). **Bitcoin Price Prediction Using Machine Learning**. *Journal of Computational and Applied Mathematics*, *365*, 112395.

- Algorithm is taken from the above paper
- My own work in the project is the additional features I engineered and applied to the models

Secondary Research Papers:

Chen, J. (2023). **Analysis of Bitcoin Price Prediction Using Machine Learning**. *Journal of Risk and Financial Management*, *16*(1), 51. **https://doi.org/10.3390/jrfm16010051**

Ortu, M., Uras, N., Conversano, C., Bartolucci, S., & Destefanis, G. (2022). **On technical trading and social media indicators for cryptocurrency price classification through deep learning**. *Expert Systems With Applications*, *198*, 116804. **https://doi.org/10.1016/j.eswa.2022.116804**