Macroeconomic Effects on Bitcoin Price Using Topological Data Analysis and Distance-to-Default Metrics

Eric Schmid

October 2024

Abstract

In this report, we explore the application of Topological Data Analysis (TDA) and financial risk metrics such as the Merton Distance-to-Default (DTD), adapted slightly here, in predicting Bitcoin price deciles. Using daily data from Yahoo Finance and FRED, we build a machine learning model that integrates both financial and geometric features. In this study, the target variable is the 30-day moving average of Bitcoin's price, calculated with a 15-day lookahead window. This metric captures smoothed price behavior while projecting price movements 15 days into the future. By using the 30-day average, the analysis reduces short-term volatility, allowing for more stable predictions. The model is trained using XGBoost with hyperparameter optimization through Bayesian search, achieving an accuracy of 80.69% and an AUC score of 0.531. To better understand the model's performance, we evaluate entropy per class and plot the confusion matrix, as well as predicted vs actual values over time.

Repository

For full access to the code and related files for this project, please visit the GitHub repository at:

`https://github.com/ericschmid-uchicago/macroeconomic-trends-on-bitcoin`

1 Introduction and Literature Review

Bitcoin is a decentralized digital currency whose market value is highly volatile, creating both opportunities and risks for investors. This volatility makes Bitcoin an ideal candidate for mathematical modeling. Traditional models, such as the Merton DTD model, are commonly used in finance to estimate the distance to default for companies, providing insight into financial risk.

Topological Data Analysis (TDA) is a novel approach that has gained traction in recent years. TDA captures the underlying geometric and topological structures in data, providing distinctive insights that conventional approaches might overlook. Research has shown that TDA can be useful for analyzing high-dimensional time-series data, such as stock market prices and cryptocurrency data [2]. In this paper, we apply TDA to Bitcoin data to extract topological features and use them, along with traditional financial metrics like Treasury yields and Gross Debt, to improve predictions of Bitcoin price changes.

1.1 Bitcoin as a Financial Asset

Bitcoin's decentralized nature and lack of intrinsic value have led to substantial price fluctuations. Its adoption as an investment medium has led to a growing body of literature focused on understanding and predicting its price movements. However, traditional financial indicators such as Treasury yields, inflation rates, and debt levels may still influence Bitcoin, as they reflect the broader economic environment.

1.2 Topological Data Analysis in Finance

TDA has been applied in various fields, including neuroscience, biology, and recently, finance. Persistent homology, one of the key tools in TDA, captures the multi-scale topological features of data by identifying clusters, loops, and voids. These features, called persistence diagrams, are then used to quantify the "shape" of data and provide additional dimensions for analysis. In financial time-series data, TDA can capture subtle, persistent patterns that other methods might overlook.

2 Data Collection and Feature Engineering

2.1 Bitcoin Data from Yahoo Finance

For this project, we used Bitcoin historical price data fetched from Yahoo Finance. The dataset spans from 2014 to 2024, providing sufficient granularity and time depth for meaningful analysis. The dataset includes daily "closing" prices, which serve as the basis for our calculations. The following code fetches and preprocesses the Bitcoin data, ensuring that any timezone information is removed:

```
btc = yf.download("BTC-USD", start="2014-01-01")["Adj Close"]
btc = btc.reset_index()
btc['Date'] = pd.to_datetime(btc['Date']).dt.tz_localize(None)
btc.columns = ['Date', 'BTC']
```

This ensures consistency across the dataset for the next steps.

2.2 FRED Economic Indicators

To capture the broader economic context, we obtained financial data from the Federal Reserve Economic Data (FRED) API. The two primary economic indicators used were the 10-Year Treasury Yield and the U.S. Gross Federal Debt. These metrics reflect the cost of borrowing and the nation's financial obligations, respectively, both of which could influence investor behavior in cryptocurrency markets. These factors were hypothesized to have an impact on Bitcoin's price due to their reflection of macroeconomic conditions:

```
fred = Fred(api_key=fred_api_key)
treasury_10y = fred.get_series("DGS10", observation_start="2014-01-01")
gross_debt = fred.get_series("GFDEBTN", observation_start="2014-01-01")
```

3 Mathematical Models and Methods

3.1 Merton's DTD Model for Bitcoin

The Merton DTD model, traditionally used in corporate finance to assess a company's risk of default, was adapted to the cryptocurrency market in this project. The model calculates the distance to default (DTD) for Bitcoin using the following formula:

$$DTD = \frac{\ln(A/D) + (r - 0.5\sigma^2)T}{\sigma\sqrt{T}}$$

where:

- A is the "asset price" (Bitcoin price),
- D is the "debt" (Gross Federal Debt),
- r is the risk-free interest rate (Treasury yield),
- σ is the volatility of the asset (Bitcoin),
- T is the time horizon.

This metric serves as a proxy for Bitcoin's financial stability and is used as one of the features in our machine learning model.

The Merton Distance-to-Default model is used to assess the risk of an asset's value falling below its liabilities. The DTD is calculated as follows:

```
mert['vol_BTC'] = mert['BTC'].pct_change().rolling(window=30, min_periods=30)
                  .std().shift(1) * np.sqrt(252)
A = mert['BTC'].shift(1)
D = mert['Gross_Debt'].shift(1)
r = mert['Treasury_10Y'].shift(1) / 100
T = 1  # Time horizon in years
mert['btc_d2'] = (np.log(A / D) + (r - 0.5 * mert['vol_BTC'] ** 2) * T) /
                 (mert['vol_BTC'] * np.sqrt(T))
mert['btc_d2'] = mert['btc_d2'].fillna(0)
```

3.2 Topological Data Analysis (TDA)

We performed TDA on the DTD data using Vietoris-Rips persistence. By applying TDA, we captured multi-scale geometric structures in the DTD time series, which are summarized using the bottleneck distance:

```
VR_persistence = VietorisRipsPersistence(metric="euclidean",
                                          homology_dimensions=[0, 1])
persistence_diagrams = VR_persistence.fit_transform(d2_windows)
amplitude = Amplitude(metric='bottleneck')
tda_features = amplitude.fit_transform(persistence_diagrams)
```

These features were padded and aligned with the dataset.

The TDA methodology focuses on transforming time-series data into topological features. For this project, we used the 'gtda' library to compute the persistence diagrams of Bitcoin price data. Persistence diagrams are a graphical representation of the homological features, such as connected components and loops, which persist across multiple scales in the data.

The persistence diagrams are then transformed into numerical features (e.g., persistence entropy and amplitude) that can be fed into the machine learning model. These features help capture long-term trends and structures in the Bitcoin price data that traditional statistical methods might miss.

4 Machine Learning: XGBoost Model

The XGBoost model was chosen due to its efficiency and high performance on structured datasets. It is a gradient boosting algorithm that builds a series of decision trees, where each tree corrects the errors of the previous one.

We split the dataset into training and test sets using a time-series-aware split. The dataset was divided 80% for training and 20% for testing, ensuring that no data leakage occurred. The model used the following features:

- **Bitcoin price:** the daily closing price.
- **Merton DTD:** the calculated distance to default.
- **TDA features:** derived from persistence diagrams.
- **Treasury Yield:** the 10-Year Treasury Yield.
- **Gross Debt:** the U.S. Gross Federal Debt.

5 Target Variable: 30-Day Moving Average with 15-Day Lookahead

The lookahead window anticipates future price changes, which is especially valuable in highly volatile markets like Bitcoin. This approach offers a balanced method for forecasting near-term trends while mitigating noise from daily fluctuations.

6 Model Training and Optimization

6.1 Model Setup

We used the XGBoost model to classify Bitcoin price changes into deciles (0, 1, 2). The model was trained with Bayesian search over the following parameter space:

```
search_space = {
    'n_estimators': Integer(50, 300),
    'max_depth': Integer(3, 10),
    'learning_rate': Real(0.01, 0.3, 'log-uniform'),
    'subsample': Real(0.6, 1.0),
    'colsample_bytree': Real(0.6, 1.0),
    'lambda': Real(0.01, 10.0, 'log-uniform'),  # L2 regularization
    'alpha': Real(0.01, 10.0, 'log-uniform')    # L1 regularization
}
```

Bayesian search with Gaussian Processes was used to optimize the hyperparameters.

6.2 Evaluation Metrics

The model's performance was evaluated using AUC and accuracy scores. Additionally, confusion matrices and entropy plots were generated to understand how well the model performed on each class.

7 Results

7.1 Confusion Matrix

The matrix compares the actual classes (on the vertical axis) against the predicted classes (on the horizontal axis). A perfect classification model would have all non-zero values along the diagonal of the matrix, indicating that every predicted class matches the actual class. However, off-diagonal elements indicate misclassifications, where the model has predicted the wrong class.

Class 0: The model performs particularly well for class 0, with 693 correct predictions. However, there are still 84 instances where the model incorrectly classified class 0 as class 1, and 42 instances where class 0 was misclassified as class 2. This suggests that the model occasionally struggles to differentiate class 0 from adjacent classes, particularly class 1. Despite this, the majority of predictions for class 0 are correct.

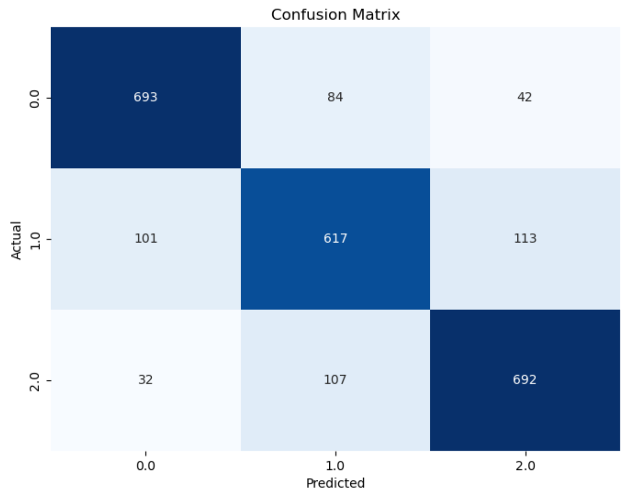

Figure 1: Confusion Matrix for Predicted vs Actual Classes

Class 1: Class 1 poses more challenges for the model. While 617 instances were correctly classified, there were 101 cases misclassified as class 0, and 113 misclassified as class 2. The confusion between class 1 and class 2 indicates that the model has difficulty distinguishing these two classes. This could be due to similarities in price movement patterns between these deciles, leading to greater overlap in the feature space.

Class 2: For class 2, the model correctly predicted 692 instances, but 107 cases were incorrectly predicted as class 1, and 32 instances as class 0. The confusion between classes 1 and 2 further highlights the difficulty the model faces in clearly differentiating between these two deciles. However, class 2 is relatively well separated from class 0, with a low misclassification rate of just 32 instances.

Overall, while the model achieves a high level of accuracy, the confusion matrix highlights areas where further improvements can be made. Enhancing the feature set or using more advanced techniques, such as additional regularization or ensemble models, could help address these misclassification issues.

7.2 Predicted vs Actual Over Time

Figure 2 plots the predicted class labels against the actual class labels over time. The model predictions closely follow the actual class changes, though there are instances of mismatch, particularly during periods of high volatility.

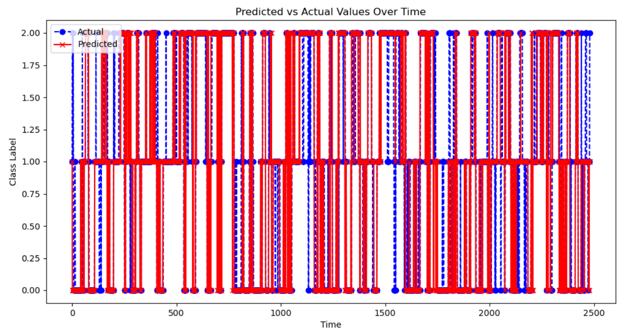

Figure 2: Predicted vs Actual Values Over Time

7.3 Entropy per Class

In Figure 3, the mean entropy values per class are presented. Higher entropy for class 1.0 indicates that the model is less confident when predicting this class, suggesting a need for further refinement.

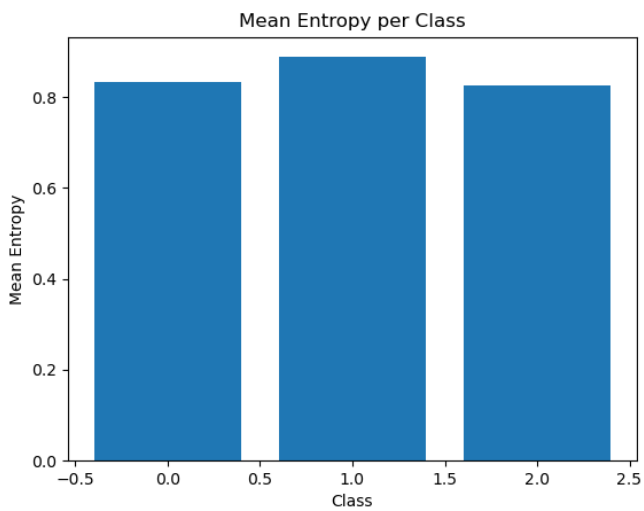

Figure 3: Mean Entropy per Class

8 Conclusion

The integration of Topological Data Analysis with financial metrics, such as Distance-to-Default, demonstrates the potential to improve Bitcoin price forecasting. While the model achieves reasonable accuracy, further refinements in feature engineering and regularization could lead to higher confidence and reduced entropy, particularly in class 1.0. Future research could explore more sophisticated geometric descriptors and the inclusion of additional macroeconomic factors to improve predictive performance.

References

[1] Merton, R. C. (1974). On the Pricing of Corporate Debt: The Risk Structure of Interest Rates. *Journal of Finance*, 29(2), 449-470.

[2] S. W. Akingbade, M. Gidea, M. Manzi, and V. Nateghi, *Why Topological Data Analysis Detects Financial Bubbles?*, arXiv preprint arXiv:2304.06877, 2023. Available at: `https://arxiv.org/abs/2304.06877`.

[3] Scikit-TDA Developers (2023). *gtda: A Python Library for Topological Data Analysis*.