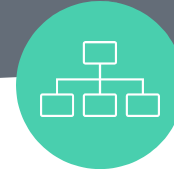
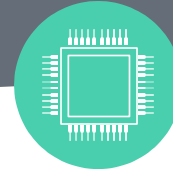




Bitcoin Prediction



By Greg Stevenson
Kyle Hagan
Robel Gebremeskel

Motivation and Summary

The goal of our project was to create a machine learning model that effectively predicted the value of Bitcoin using a couple different machine learning libraries.

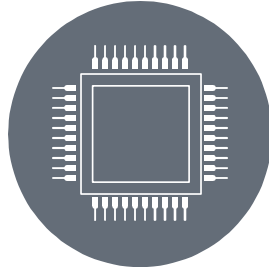
Model Summary

Kernel



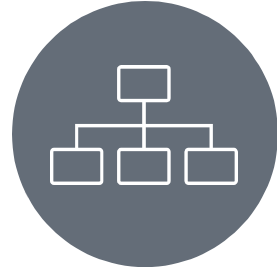
Research Different ML Models

Find different models
that would be able to
predict BTC price



Use yfinance API


Gather historical data
from the API of
yfinance



Create new features

EMA9, SMA5,
SMA10, SMA15,
SMA30

Data Preparation

- Used yfinance library to pull Bitcoin data
 - Dropped the Adj Close column since it didn't apply to BTC
 - Added features EMA9, SMA5, SMA10, SMA15, SMA30, and Percent Change
 - Created a date defined test and training period
 - Removed NaN from Dataframe
 - Sliced Training data and Test Data
- 
- A teal-colored decorative bar with a diagonal gradient, starting from the bottom left and extending towards the top right, located at the bottom of the slide.

Kernel Ridge Model

What is it?

- KernelRidge combines Ridge regression and classification with the kernel trick.

Why use it?


- Fitting **KernelRidge** is faster than some other models with data sets with less than 1000 samples

Model Evaluation & Training

Kernel Ridge Performance:

- Mean Absolute Error - 134.66
- Mean Squared Error - 187.08
- R-Squared - 0.99


XGBoost Model

- Data preparation
 - Feature Selection
 - Increased features using rolling window 3d, 7d, 30d
 - Data Splitting
 - Splitting the data into training and testing sets to evaluate model's performance.
 - Included hyperparameter to optimize our model
- 
- A teal-colored decorative bar with a gradient effect, starting from a solid teal at the bottom and fading into white at the top, located at the bottom of the slide.

XGBoost Model Evaluation

- Data Visualization
 - Visualization of the data to gain a better understanding of model

Model accuracy evaluated with key metrics

- Mean Absolute Error – 2540.45
 - Mean Squared Error - 3132.84
 - R-Squared – 0.95
- 
- A teal-colored decorative bar with a gradient, starting from a solid teal on the left and fading into white on the right, located at the bottom of the slide.

Conclusion

Performance comparison:

- Kernel Ridge Regression Model appeared to have more accurate prediction compared to the XGBoost model

Future applications:

- As the sample size grows, using XGBoost may be a more suitable library
- Further tune existing KR model which may include adding/removing features
 - May also consider sentiment analysis

Challenges

- Applying Kernel Ridge Regression forecast into the future beyond the next day
 - Prophet may solve forecasting challenges with time series data
- Properly indexing starting date with useful intervals
 - There is an 'interval' parameter with yfinance which can be used to look at a number of different time intervals
- Understanding all the parameters included with XGBoost
 - Simply requires additional time and research

Q&A