Crypto Trend Predictor Optimization

Changes Made in 11/20/2023 Update

- Today we focused on optimzation of our code.

1. Modularization and Functionality Separation

- **Before Optimization:**
  - The entire code was written in a single script with limited separation of concerns.
  - Data loading, feature engineering, model training, and prediction were all in one place.
- **After Optimization:**
  - The code has been modularized into separate functions, improving readability and reusability.
  - Different aspects such as data processing, visualization, feature engineering, and model evaluation are now handled in distinct functions.

2. Data Processing and Visualization

- **Before Optimization:**
  - Data loading, sorting, and exploratory data analysis were combined in a single script.
  - The code lacked clear visualization functions.
- **After Optimization:**
  - The `load_data` function loads data, sorts it, and resets the index.
  - The `visualize_data` function generates a plot for better understanding of BTC close prices over time.

3. Feature Engineering

- **Before Optimization:**
  - Feature engineering was embedded within the main script.
  - Lagged returns, moving averages, and other features were created directly.
- **After Optimization:**
  - The `feature_engineering` function is dedicated to creating new features, providing a cleaner and more modular approach.

4. Model Training and Evaluation

- **Before Optimization:**
  - Model training, testing, and evaluation were performed directly in the main script.
  - Limited flexibility in trying different models.

- **After Optimization:**
  - The `train_model` function allows for training various regression models.
  - The `evaluate_model` and `evaluate_model_cv` functions handle model evaluation, providing flexibility for cross-validation.