Bitcoin Price Prediction

Using Chainlet Data and Ridge Regression

Introduction:

This report details the process and results of predicting Bitcoin prices for December 2017 using chainlet data and historical price information. The goal was to build a predictive model, evaluate its performance, and explain its behavior using SHAP analysis.

Data Preprocessing:

- **Data Loading:** The datasets pricedBitcoin2009-2018.csv, AmoChainletsInTime.txt, and OccChainletsInTime.txt were loaded using pandas.
- **Date Handling:** The 'date' column in pricedBitcoin2009-2018.csv was converted to datetime objects. Year and day columns were extracted.
- **Data Merging:** The chainlet data (amount and occurrence) was merged with the price data using 'year' and 'day' as keys.
- **Lagged Price Feature:** A lagged price feature ('price_lag1') was created to represent the previous day's price.
- **Data Filtering:** The data was filtered to include only the year 2017.
- **Feature Selection:** Specific chainlet features ('1:1_amo' and '1:1_occ') were selected, and unnecessary columns were dropped.
- **Data Splitting:** The data was split into training (January-November 2017) and testing (December 2017) sets.

Model Selection and Training:

- A Ridge Regression model was chosen for its simplicity and effectiveness in handling linear relationships with regularization.
- The model was trained using the training data, with an alpha value of 1.0.

Evaluation and Results:

- The model's performance was evaluated using Root Mean Squared Error (RMSE).
- The RMSE value was: 1110.05
- The predictions for each day of December 2017 were saved to a CSV file (price_predictions_ridge.csv).
- A plot comparing the actual and predicted Bitcoin prices was generated.

Explanation Findings (SHAP Analysis):

- SHAP decision plots were used to explain the model's predictions for the first five days of December 2017.
- The plots showed the progression of the model's prediction as features were considered, highlighting the impact of each feature.
- On 2017-12-01, the most influential feature was 'price_lag1', which significantly increased the predicted price by approximately 7577 USD. The '1:1_amo' and '1:1_occ' chainlet features also had a positive impact, though to a lesser extent.
- On 2017-12-03, 'price_lag1' remained the dominant feature, positively affecting the prediction by approximately 8517 USD. The '1:1_amo' feature also contributed positively, while '1:1_occ' had a minimal effect.
- On 2017-12-05, 'price_lag1' continued to be the most influential, increasing the prediction by approximately 9040 USD. The '1:1_occ' feature had a moderate positive impact, while '1:1_amo' had a smaller effect.

Challenges and Discussion:

- **Data Integration Complexity:**
  - Merging the chainlet data (amount and occurrence) with the price data required careful handling of date alignment. The "day of year" representation, while necessary, introduced potential complexities in ensuring accurate synchronization between datasets.
  - The different file formats (.csv and .txt) added a layer of complexity to the initial data loading and parsing.
- **Feature Selection and Dimensionality Reduction:**
  - The chainlet datasets contained a large number of features (400 chainlet types), which posed a challenge in terms of feature selection and potential overfitting.

  - Deciding which chainlet features to include in the model was a critical step. We chose to focus on '1:1_amo' and '1:1_occ' as a starting point, but this decision could significantly impact the model's performance.
  - Dropping the year and day features, as per the assignment constraints, increased the difficulty of the time series predictions.
- **Lagged Feature Creation:**
  - Creating the lagged price feature ('price_lag1') introduced missing values in the first row of the DataFrame, requiring careful handling (dropping the row).
  - Deciding on the correct amount of lagged features to include, was a challenge. Only one lagged feature was used, but more could have been tried.
- **Time Series Data Splitting:**
  - Splitting the time series data into training and testing sets required careful consideration to avoid data leakage. We chose a chronological split (January-November 2017 for training, December 2017 for testing) to maintain the temporal order of the data.
- **Model Selection and Parameter Tuning:**
  - While Ridge Regression was chosen for its simplicity and regularization, exploring other models (e.g., Random Forest, Gradient Boosting) could have potentially yielded better results.
  - The alpha parameter in Ridge Regression was set to 1.0, but further tuning could have optimized the model's performance.
- **Potential Data Noise:**
  - The chainlet data, while potentially informative, might contain noise or irrelevant information that could negatively impact the model's predictions.

Conclusion:

The Ridge Regression model demonstrated moderate prediction success in predicting Bitcoin prices for December 2017. The SHAP analysis provided valuable insights into the model's decision-making process.

CSV File:

- A separate CSV file (price_predictions_ridge.csv) containing the predicted prices for each day of December 2017 is included with this report.